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

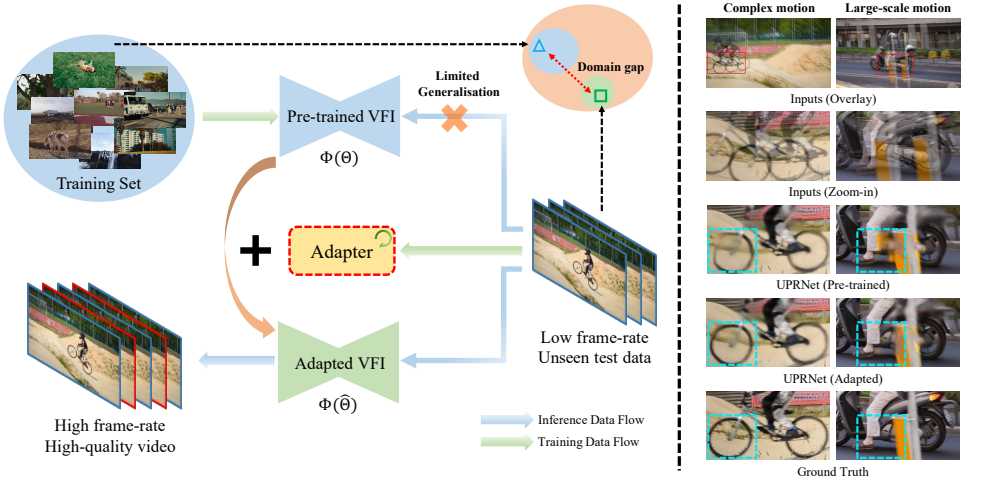
Abstract

Video frame interpolation (VFI) is a challenging task that aims to generate intermediate frames between two consecutive frames in a video. Existing learning-based VFI methods have achieved great success, but they still suffer from limited generalization ability due to the limited motion distribution of training datasets. In this paper, we propose a novel optimization-based VFI method that can adapt to unseen motions at test time. Our method is based on a *cycle-consistency adaptation* strategy that leverages the motion characteristics among video frames. We also introduce a lightweight *adapter* that can be inserted into the motion estimation module of existing pre-trained VFI models to improve the efficiency of adaptation. Extensive experiments on various benchmarks demonstrate that our method can boost the performance of two-frame VFI models, outperforming the existing state-of-the-art methods, even those that use extra input frames. Project page: https://haoningwu3639.github.io/VFI_Adapter_Webpage/

1 Introduction

Video frame interpolation (VFI) is a technique that increases the temporal resolution of a video by synthesizing intermediate frames between existing frames. This results in smoother transitions between frames, which can improve the overall quality of the video. VFI has a wide range of applications, including video compression [35, 47], slow-motion generation [25] and novel-view synthesis [8, 17], *etc*. In the literature, existing VFI approaches can be roughly divided into two categories: flow-agnostic and flow-based frame synthesis. Flow-agnostic approaches do not use optical flow to compensate for motion between frames, instead, they typically use a combination of adaptive convolution kernel and interpolation

†: Corresponding author.



(a) Overall pipeline of our proposed optimisation-based video frame interpolation (b) Comparison on critical cases

Figure 1. High-level idea overview. (a) To address the generalisation challenge of VFI models due to domain gap on unseen data, we propose the optimisation-based video frame interpolation. By performing test-time motion adaptation on our proposed lightweight *adapter*, we achieve the generalization of VFI models across different video scenarios and subsequently boost their performance. (b) Visual comparison on the cases with complex and large-scale motions from DAVIS [36] dataset. Our method assists VFI models in generalising to diverse scenarios and synthesizing high-quality frames with clearer structures and fewer distortions.

techniques. Flow-based approaches, on the other hand, do use optical flow to estimate the motion between frames. This information is then used to synthesize the intermediate frames. Recently, thanks to the rapid progress in optical flow estimation [43, 45], flow-based VFI approaches [2, 9, 14, 15, 29] have become the dominant approach in the field.

Existing flow-based frame interpolation models are learning-based, that usually follow a similar pipeline: extract visual features from the input frames, estimate optical flows between the reference and target frame to be synthesized, warp the input frames and their contextual features based on the estimated flows, then finally synthesize the intermediate frame based on the aligned visual features. Under such design, various model architectures have been explored, such as constructing multi-scale pyramids [11, 15, 19] and designing vision transformer [27, 40]. Additionally, the incorporation of extra input information, such as depth [1], or more adjacent frames [18, 23, 40, 49], have also been explored as potential solutions. Despite training such meticulously-designed models on a large amount of videos, generalisation towards real videos with complex and large-scale motions still remains challenging.

Unlike existing approaches on architecture design, we explore an alternative direction for boosting models' performance at inference time, *i.e.* optimisation-based frame interpolation via test-time motion adaptation. Test-time adaptation has been proven effective in enhancing the performance of models in specific scenarios, as demonstrated in numerous computer vision tasks, such as image super-resolution [10, 28, 41] and image deblurring [4], *etc*, it remains unexplored in the domain of video frame interpolation. To this end, we devise a novel strategy suitable for video frame interpolation, namely *cycle-consistency adaptation*. The key idea is to construct triplet samples with consecutive frames from low frame-rate videos during test time and optimise model parameters on each video sequence by leveraging the inter-frame consistency. Considering that the adaptation process necessitates considerable

inference time, we further propose a simple, yet effective *adapter* that can be injected into the existing motion estimation module of VFI, enabling motion refinement with less than 4% trainable parameters comparing to the original model.

To summarise, we make the following contributions in this paper: (i) to improve the generalisation ability of existing VFI models and boost their performance, we propose an optimisation-based motion adaptation strategy based on cycle-consistency, that allows to tune the model on each test video sequence at inference time; (ii) to address the drawback of high time cost associated with test-time adaptation, we design a simple, yet effective plug-in *adapter* to refine the motion flow estimated by VFI models, with minimal tuning cost; (iii) we experiment on various models and benchmarks, demonstrating that our optimisation-based method can always boost the performance of existing two-frame VFI models, even outperforming approaches with extra inputs.

2 Related Work

Video Frame Interpolation. Video frame interpolation (VFI) is a long-standing computer vision research topic and has been widely studied. The recent literature on training deep neural networks has demonstrated extraordinary performance for video frame interpolation. Depending on whether optical flow is used, VFI methods can be broadly classified into two categories: flow-agnostic [3, 6, 18, 21, 31, 32, 39] and flow-based ones [2, 9, 14, 15, 16, 29, 30, 33, 34]. With the rapid development of optical flow estimation algorithms [12, 13, 43, 45], flow-based approaches have taken the dominant position, which typically employ optical flow to warp visual features of adjacent frames to synthesize intermediate frame, hence the quality of the generated frame is highly affected by the accuracy of motion estimation. Various strategies have been explored to improve the performance of flow-based methods. These include exploring depth information for occlusion reasoning [1], guiding the learning of motion estimation via knowledge distillation [11, 19], designing efficient architectures for high-resolution videos with relatively large motion [37, 42], utilizing the long-range dependency modeling capability of transformer for processing extensive motion [27, 40], and making full use of multiple adjacent frames for complex motion modeling [18, 23, 40, 49].

In contrast to the aforementioned learning-based methods that aim to enhance VFI model generalisation and performance by modifying model architecture or incorporating extra input information, we consider an optimisation-based video frame interpolation that adapts pre-trained VFI models to the motion patterns in different video sequences at inference time.

Cycle Consistency. The idea of cycle consistency has been widely explored in various self-supervised methods, such as image representation and correspondence learning [7, 20, 46, 48, 53]. For low-level vision tasks, CycleGAN [54] utilizes cycle-consistency loss to constrain the training process of generative adversarial networks on image2image translation task. ARIS [52] exploits cycle-consistency constraint to augment the models' ability for arbitrary super-resolution. CyclicGen [24] and Reda *et al*. [38] learn video frame interpolation in an unsupervised manner with the proposed cycle-consistency loss on general video data.

Test-time Adaptation. Test-time adaptation manages to adapt the trained model to test data distribution for performance improvement, and has been successful in tasks such as classification [44] and pose estimation [22], *etc*. Moreover, the idea has also been widely employed in low-level vision, such as super-resolution [10, 28, 41], in order to improve the generalisation ability of models on various data. For video frame interpolation, SAVFI [8] utilizes a

meta-learning framework for training models to achieve scene-adaptive frame synthesis.

In this paper, we exploit *cycle-consistency* for test-time motion adaptation in video frame interpolation task, optimising motion for test video sequences on-the-fly, thus improving performance steadily. Moreover, a lightweight yet effective plug-in *adapter* has been proposed to improve the efficiency of VFI test-time adaptation.

3 Methods

In this section, we start by introducing the considered problem scenario, *e.g.*, learning-based and our proposed optimisation-based video frame interpolation (Sec. 3.1). Subsequently, we detail our proposed motion adaptation strategy that is suitable for video frame interpolation, namely *cycle-consistency adaptation* (Sec. 3.2). Lastly, to further improve the efficiency of test-time adaptation, we present a lightweight *adapter* that can serve as a plug-in module for better motion estimation in VFI models (Sec. 3.3).

3.1 Problem Scenario

Given a low frame-rate input video, the goal of video frame interpolation is to synthesize intermediate frame between two or multiple adjacent frames, ending up of high-frame videos with smoother motion. In a learning-based VFI model, it typically takes two consecutive frames ($\mathcal{I}_{i-1}$ and $\mathcal{I}_{i+1}$) as input and outputs one single intermediate frame ($\mathcal{I}_i$) between them, the model's parameters are learnt by minimizing the empirical risk on a training set:

$$\mathcal{L}_{\mathcal{D}}(\Theta) = \mathbb{E}_{\mathcal{D}}(||\hat{\mathcal{I}}_i - \mathcal{I}_i||) \qquad \text{where } \hat{\mathcal{I}}_i = \Phi(\mathcal{I}_{i-1}, \mathcal{I}_{i+1}; \Theta) \tag{1}$$

$\Phi(\cdot)$ refers to the video frame interpolation model, Θ denotes the parameters to be learnt on a large-scale training set ($\mathcal{D}$), and $\{\hat{\mathcal{I}}_i\}$ denote the predicted intermediate frames. At inference time, the model is expected to generalise towards unseen videos. However, in practise, these models can sometimes be fragile on cases with diverse and complex motions.

In this paper, we consider to improve the model's efficacy, with optimisation-based video frame interpolation via test-time motion adaptation:

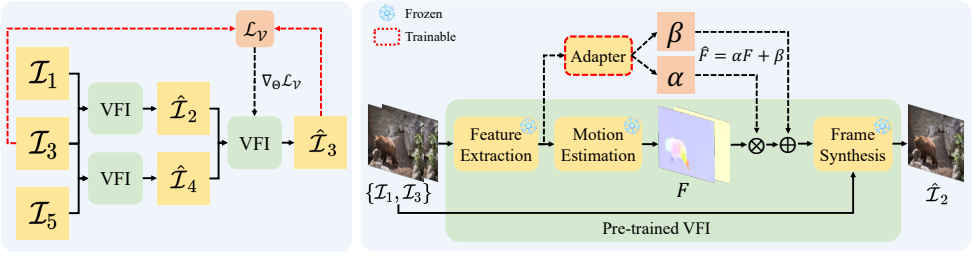
$$\hat{\Theta}_{\mathcal{V}} = \arg\min_{\Theta} \mathcal{L}_{\mathcal{V}}(\Phi, \mathcal{V}; \Theta) \tag{2}$$

where $\Phi(\cdot)$ denotes a pre-trained VFI model with parameters Θ, and we aim to further optimise its parameters to boost the performance on one given test video sequence ($\mathcal{V}$). In the following section, we aim to answer the core question: **how can we design the objective function ($\mathcal{L}_{\mathcal{V}}$), given only low frame-rate videos are presented at inference time ?**

3.2 Cycle-Consistency Adaptation

At inference time, we construct a series of triplet samples from the given test video sequence, each consists of three consecutive frames, for example, $\mathcal{D}_1 = \{\mathcal{I}_1, \mathcal{I}_3, \mathcal{I}_5\}$, $\mathcal{D}_2 = \{\mathcal{I}_3, \mathcal{I}_5, \mathcal{I}_7\}$, *etc*. Our goal is to optimise the model's parameters using $\mathcal{D}_1$, and to boost the performance on synthesizing intermediate frames $\{\hat{\mathcal{I}}_2, \hat{\mathcal{I}}_4\}$ by exploiting the cycle-consistency constraint.

As detailed in Figure 2 (a), taking the triplet $\mathcal{D}_1 = \{\mathcal{I}_1, \mathcal{I}_3, \mathcal{I}_5\}$ for demonstration, we can generate the intermediate frames with a pre-trained off-the-shelf VFI model on input



(a) Cycle-consistency adaptation strategy (b) Plug-in adapter module

Figure 2. Proposed *cycle-consistency adaptation* strategy and plug-in *adapter* module for efficient test-time adaptation. (a) *Cycle-consistency adaptation* first synthesizes intermediate frames between each two input frames and reuses them to interpolate the target frame to calculate cycle-loss, which fully utilizes the consistency within video sequences. (b) To improve efficiency, we freeze all the parameters of pre-trained VFI models and solely optimise the proposed plug-in *adapter*, which predicts a set of parameters $\{\alpha, \beta\}$ based on the extracted visual features. The pixel-wise weights α and biases β are used for rectifying the estimated flow to fit each video sequence.

consecutive video frames,

$$\hat{\mathcal{I}}_2 = \Phi(\mathcal{I}_1, \mathcal{I}_3; \Theta) \qquad \hat{\mathcal{I}}_4 = \Phi(\mathcal{I}_3, \mathcal{I}_5; \Theta) \tag{3}$$

and reuse the synthesized frames to predict the target frame:

$$\hat{\mathcal{I}}_3 = \Phi(\hat{\mathcal{I}}_2, \hat{\mathcal{I}}_4; \Theta) \tag{4}$$

The model's parameters Θ can be updated according to:

$$\Theta \leftarrow \Theta - \eta \nabla_\Theta \mathcal{L}_\mathcal{V}(\hat{\mathcal{I}}_3, \mathcal{I}_3) \quad \text{where } \mathcal{L}_\mathcal{V}(\Theta) = ||\hat{\mathcal{I}}_3 - \mathcal{I}_3|| \tag{5}$$

η denotes the adaptation learning rate. This cycle-consistency adaptation strategy enables the model to make full use of inter-frame consistency, thereby acquiring motion patterns that are more suitable for testing scenarios and achieving stable performance improvements.

3.3 Lightweight Motion Adaptation with Plug-in Adapter

Optimising the entire model for test-time motion adaptation incurs computation overhead, here, we further propose a simple, lightweight plug-in *adapter* module that can be inserted into the motion estimation module of existing pre-trained VFI models, requiring minimal tuning to boost the performance. As depicted in Figure 2 (b), we freeze all parameters of the pre-trained VFI model, *e.g.*, feature extraction, motion estimation and frame synthesis modules, then incorporate the proposed adapter into the motion estimation module, which takes visual features extracted from adjacent frames as input, and predicts a set of parameters $\{\alpha, \beta\}$ to adjust motion estimation per video sequence.

To be specific, the adapter reuses the convolutional features from motion estimation module, and subsequently employs a 1×1 Convolution layer to transform the feature map into pixel-wise weights α and biases β:

$$\hat{F} = \alpha F + \beta, \text{ where } \{\alpha, \beta\} = \text{Conv}(\Psi(M)) \tag{6}$$

M represents the extracted visual feature map and Ψ denotes the resused convolutional layers, then the predicted $\{\alpha, \beta\}$ are used to modify the estimated motion flow F. During test-time adaptation, we only finetune the parameters in adapter, effectively refining the estimated motion flow and boosting the performance of video frame interpolation model.

4 Experiments

In this paper, we start from a series of VFI models that have been pre-trained on Vimeo90K-Triplet dataset [50], then train our proposed *adapter* module on the same dataset. For test-time motion adaptation, the model is further optimised on each testing sequence from three benchmark datasets, including Vimeo90K-Septuplet [50], DAVIS [36], and SNU-FILM [6].

Training Set. Vimeo90K-Triplet [50] training set comprises 51,312 triplets, wherein each triplet contains three consecutive video frames with a spatial resolution of 448×256 pixels.

Testing Set. Vimeo90K-Septuplet [50] dataset encompasses 7,824 seven-frame sequences for testing, with a fixed spatial resolution of 448×256 pixels. DAVIS [36] is a typically high-quality video segmentation dataset with a fixed resolution of 854×480 pixels. Following FLAVR [18] and VFIT [40], we also report performance on 2,847 septuplet test samples from DAVIS. SNU-FILM [6] dataset contains 1,240 triplets, with a predominant resolution of approximately 1280×720 pixels. It comprises of four categories with ascending motion scales: easy, medium, hard, and extreme. We further sample the surrounding frames of the ground truth and extend each triplet into a septuplet. As a result, the easy, medium, and hard categories contain 310 sequences, while the extreme category contains 234 sequences. Among these septuplets in the above three benchmarks, the four odd frames compose the input video sequence of the VFI model, as described in Sec. 3.1. And the intermediate frame is regarded as the ground truth for the frame to be interpolated in our experiments.

Evaluation Metrics. Following common practise, we report Peak Signal-to-Noise Ratio (PSNR) and Structural-Similarity-Image-Metric (SSIM) on the RGB channel of the target interpolated frame for the three benchmark datasets.

Training Details. To start with, we freeze the parameters of the pre-trained VFI models and only train our proposed plug-in adapter on Vimeo90K for 30 epochs with a batch size of 16. We randomly crop 256×256 patches and augment the training data using horizontal and vertical flipping, temporal order reversing and RGB channel flipping. We use the AdamW [26] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$, and the learning rate is gradually reduced from 3×10^{-4} to 3×10^{-5} using cosine annealing during the whole training process. Considering the high resolution of some video data, optimisation-based VFI is performed on one 40G NVIDIA A100 GPU. During test-time adaptation for the full model, we use a fixed learning rate of 1×10^{-5} and calculate L1 loss to fine-tune parameters of VFI models. As for adapter-boosted models, since other parameters have been frozen, we adapt the plug-in adapter module to each test sequence with a larger learning rate of 1×10^{-4} for IFRNet [19], 3×10^{-4} for UPRNet [15], and 1×10^{-3} for RIFE [11], respectively.

5 Results

In this section, we start by providing experimental results for comparison with existing state-of-the-art approaches (Sec. 5.1), showing the effectiveness of our proposed *cycle-consistency*

Methods	Adaptation		Vimeo90K [50]	DAVIS [36]	SNU-FILM [6]			
	e2e	plugin			Easy	Medium	Hard	Extreme
SepConv [32]	✗	✗	33.72 / 0.9639	26.65 / 0.8611	40.21 / 0.9909	35.45 / 0.9785	29.62 / 0.9302	24.16 / 0.8457
SepConv-ours-e2e	✓	✗	33.96 / 0.9650	26.83 / 0.8639	40.41 / 0.9911	35.71 / 0.9794	29.80 / 0.9313	24.26 / 0.8479
EDSC [3]	✗	✗	34.55 / 0.9677	26.83 / 0.8578	40.66 / 0.9915	35.77 / 0.9795	29.75 / 0.9301	24.12 / 0.8420
EDSC-ours-e2e	✓	✗	34.73 / 0.9685	26.96 / 0.8600	40.88 / 0.9917	35.98 / 0.9803	29.85 / 0.9313	24.19 / 0.8436
RIFE [11]	✗	✗	35.28 / 0.9704	27.61 / 0.8760	40.74 / 0.9916	36.18 / 0.9808	30.30 / 0.9368	24.62 / 0.8531
RIFE-ours-e2e	✓	✗	35.57 / 0.9717	27.81 / 0.8798	40.95 / 0.9918	36.58 / 0.9816	30.49 / 0.9386	24.71 / 0.8549
RIFE-ours-e2e++	✓	✗	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	✗	✓	35.56 / 0.9714	27.76 / 0.8771	40.99 / 0.9918	36.55 / 0.9825	30.48 / 0.9387	24.64 / 0.8533
IFRNet [15]	✗	✗	35.86 / 0.9729	28.03 / 0.8851	40.91 / 0.9918	36.58 / 0.9816	30.75 / 0.9403	24.85 / 0.8590
IFRNet-ours-e2e	✓	✗	36.38 / 0.9753	28.45 / 0.8936	41.21 / 0.9921	37.03 / 0.9832	31.10 / 0.9440	25.03 / 0.8634
IFRNet-ours-e2e++	✓	✗	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	✗	✓	36.01 / 0.9734	28.16 / 0.8825	41.06 / 0.9920	36.92 / 0.9834	30.88 / 0.9404	24.93 / 0.8599
UPRNet [15]	✗	✗	36.07 / 0.9735	28.38 / 0.8914	41.01 / 0.9919	36.80 / 0.9819	31.22 / 0.9422	25.39 / 0.8648
UPRNet-ours-e2e	✓	✗	36.68 / 0.9758	28.84 / 0.8997	41.31 / 0.9923	37.24 / 0.9836	31.66 / 0.9464	25.64 / 0.8699
UPRNet-ours-e2e++	✓	✗	**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	✗	✓	36.44 / 0.9751	28.69 / 0.8945	41.32 / 0.9923	37.38 / 0.9843	31.64 / 0.9448	25.69 / 0.8705
VFIformer [27]	✗	✗	36.14 / 0.9738	28.33 / 0.8898	40.93 / 0.9918	36.53 / 0.9815	30.52 / 0.9392	24.92 / 0.8580
EMA-VFI [51]	✗	✗	36.23 / 0.9740	28.07 / 0.8826	41.04 / 0.9921	36.73 / 0.9821	30.88 / 0.9400	24.92 / 0.8580
FLAVR [18]	✗	✗	36.22 / 0.9746	27.97 / 0.8806	41.09 / 0.9918	36.85 / 0.9830	31.10 / 0.9456	25.23 / 0.8676
VFIT-S [40]	✗	✗	36.42 / 0.9760	28.46 / 0.8926	41.15 / 0.9920	37.07 / 0.9845	31.39 / 0.9501	25.52 / 0.8717
VFIT-B [40]	✗	✗	36.89 / **0.9775**	28.60 / 0.8945	41.24 / 0.9921	37.06 / 0.9839	31.39 / 0.9501	25.61 / 0.8731

Table 1. Quantitative (PSNR/SSIM) comparison. We compare our boosted models to representative state-of-the-art methods on Vimeo90K [50], DAVIS [36] and SNU-FILM [6] benchmarks. Both of the optimisation approaches exhibit a substantial improvement in performance. **Note that** FLAVR [18] and VFIT [40] take **multiple frames** as input, but our boosted models can still outperform them. **RED**: best performance, BLUE: second best performance.

adaptation, in both end-to-end and plug-in adapter finetuning scenarios. After that, we conduct a series of ablation studies on the critical design choices on our adaptation strategy and the plug-in adapter module (Sec. 5.2).

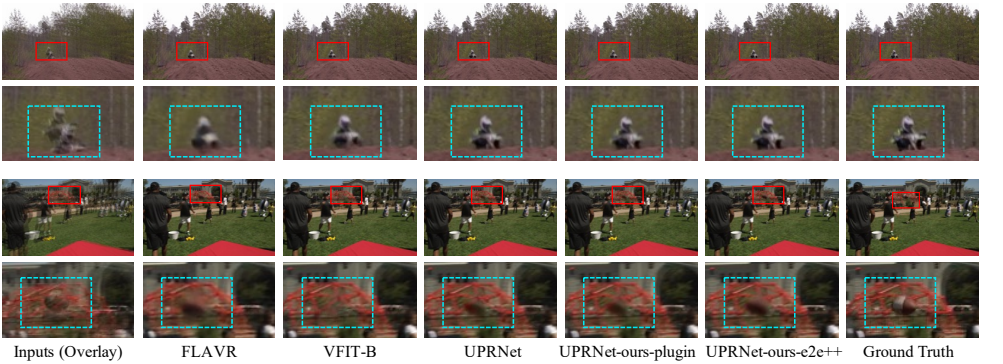
5.1 Comparison to state-of-the-art

Quantitative Results. We compare our boosted models with 9 representative learning-based models trained on Vimeo90K-Triplet [50], including flow-free ones: SepConv [32], EDSC [3] and FLAVR [18] and flow-based ones: RIFE [11], UPRNet [15] and *etc*. Among them, FLAVR [18] and VFIT [40] take four frames as input, while others only use two adjacent frames. Specifically, we consider two scenarios, namely, end-to-end finetuning (e2e), or plug-in adapter finetuning (plugin), the former optimises all parameters in the model, denoted as [model-ours-e2e], while the latter only updates adapters, denoted as [model-ours-plugin]. By default, all test-time motion adaptations are only conducted for 10-step updates, with one exception on [model-ours-e2e++], which has performed 30-step adaptation, aiming to show the performance variation with more optimisation steps.

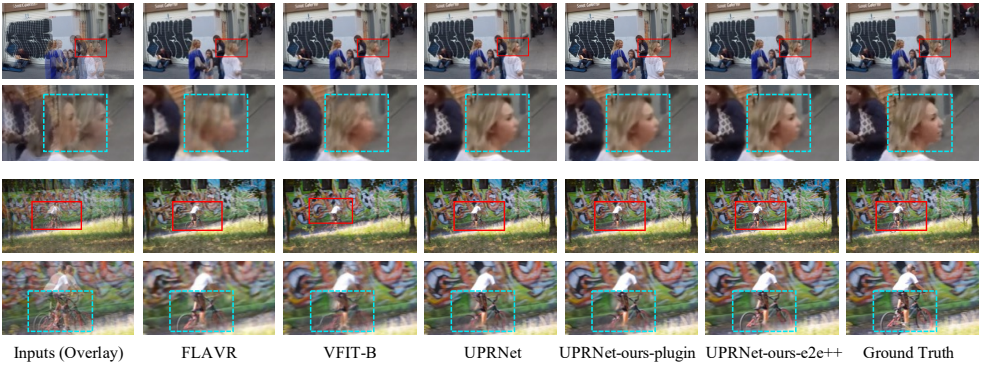
As shown in Table 1, we can draw the following three observations: (i) comparing with the off-the-shelf VFI models, our proposed *cycle-consistency adaptation* strategy with end-to-end finetuning can always bring significant PSNR performance gain on all benchmarks, that confirms the universality of our approach; (ii) the end-to-end adapted IFRNet-ours-e2e and UPRNet-ours-e2e have exhibited comparable performance to state-of-the-art methods,

such as EMA-VFI [51] and *etc*. And notably, UPRNet-ours-e2e++ with further adaptation improves **0.83dB** (36.90dB vs 36.07dB) on Vimeo90K testset and has consistently exhibited a performance gain of over **0.47dB** on other benchmarks, even outperforming the methods that take multiple frames as input, showing the effectiveness of test-time motion adaptation for unleashing the potential of pre-trained two-frame VFI models; (iii) models with the proposed plug-in *adapter* module have exhibited similar performance improvement to end-to-end finetuning, simultaneously incurring efficiency and efficacy.

Qualitative Results. We demonstrate the qualitative results in Figure 3, with the following observations: (i) comparing with existing state-of-the-art methods, the images generated by the models boosted via end-to-end adaptation and plug-in adapter present more details and have higher fidelity; (ii) our test-time optimised models generate less motion blur in the synthesized intermediate frame, indicating that the model has better adapted to the special motion characteristics in each scenario, thus improving the quality of synthesized frames.



(a) Qualitative Comparison on Vimeo90K [50]



(b) Qualitative Comparison on SNU-FILM [6] and DAVIS [36]

Figure 3. Qualitative comparison against the state-of-the-art VFI algorithms. We show visualization on Vimeo90K [50], SNU-FILM [6] and DAVIS [36] benchmarks 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.

5.2 Ablation Studies

In this section, we have conducted thorough ablation studies to quantitatively and qualitatively demonstrate the effectiveness of the proposed *cycle-consistency adaptation* strategy

Strategies	#Adaptations	SepConv [30]	EDSC [3]	RIFE [10]	IFRNet [19]	UPRNet [15]
Original	0	33.72 / 0.9639	34.55 / 0.9677	35.28 / 0.9704	35.86 / 0.9729	36.07 / 0.9735
Naïve	5	33.77 / 0.9641	34.62 / 0.9679	35.36 / 0.9708	35.95 / 0.9734	36.23 / 0.9744
	10	33.83 / 0.9644	34.69 / 0.9683	35.45 / 0.9713	35.81 / 0.9731	36.16 / 0.9747
	20	33.91 / 0.9647	34.80 / 0.9687	35.45 / 0.9715	35.03 / 0.9685	35.79 / 0.9737
	30	33.95 / 0.9648	34.85 / 0.9688	35.33 / 0.9710	34.09 / 0.9615	35.51 / 0.9721
Cycle	5	33.83 / 0.9644	34.63 / 0.9680	35.41 / 0.9710	36.14 / 0.9741	36.49 / 0.9750
	10	33.96 / 0.9650	34.73 / 0.9685	35.57 / 0.9717	36.38 / 0.9753	36.68 / 0.9758
	20	34.17 / 0.9659	34.94 / 0.9693	35.80 / 0.9728	36.60 / 0.9759	36.84 / 0.9766
	30	**34.29 / 0.9662**	**35.06 / 0.9699**	**35.93 / 0.9733**	**36.68 / 0.9760**	**36.90 / 0.9768**

Table 2. Quantitative (PSNR/SSIM) comparison of adaptation strategies. The experiments on Vimeo90K [50] dataset have shown that cycle-consistency adaptation steadily boosts VFI models by fully leveraging the inter-frame consistency to learn motion characteristics within the test sequence.

and *adapter* module from the perspectives of stability and efficiency.

Adaptation Approach. In addition to the *cycle-consistency adaptation* proposed by us, we here consider a baseline approach for test-time adaptation, which is to directly optimise the distance between $\mathcal{I}_3$ and $\hat{\mathcal{I}}_3$ synthesized with $\mathcal{I}_1$ and $\mathcal{I}_5$ as input, denoted as naïve optimisation. It is noteworthy that the inter-frame temporal distance during such adaptation is larger than that of test scenario. As shown in Table 2, we compare the two adaption strategies on five VFI methods, and have the following observations: (i) under the same adaptation steps, the performance gain of our proposed cycle-consistency adaptation is significantly higher than that of naïve adaptation; (ii) as the steps of adaptation increase, naïve adaptation may lead to a drop in performance improvement and even result in inferior performance compared to the original pre-trained models, whereas cycle-consistency adaptation can steadily boost VFI models, as it fully utilizes the inter-frame consistency to learn motion characteristics within the test video sequence.

Adaptation Cost. As mentioned in Sec. 3.3, the proposed plug-in adapter is designed to improve the efficiency of test-time motion adaptation. Here, we conduct end-to-end and plug-in adapter finetuning on three VFI models, and compare the number of parameters to be optimised and the time required for each step of adaptation. The results in Table 3 have illustrated that with the support of our proposed plug-in adapter, we can achieve a 2 times acceleration with less than 4% parameters to be optimised, while maintaining inference efficiency and similar quantitative performance improvement comparing to end-to-end finetuning. This confirms the efficiency and feasibility of our proposed plug-in adapter.

Methods	#Finetuning Parameters	Adaptation Time (ms)			Inference Time (ms)		
		Vimeo90K	DAVIS	SNU-FILM	Vimeo90K	DAVIS	SNU-FILM
RIFE-ours-e2e	10.21M	145.6	162.7	260.8	10.94	12.74	23.61
RIFE-ours-plugin	0.087M	83.13	86.84	125.4	11.79	14.67	24.79
IFRNet-ours-e2e	18.79M	107.7	196.2	403.3	18.61	25.94	55.54
IFRNet-ours-plugin	0.676M	39.08	73.79	158.1	19.11	29.32	61.58
UPRNet-ours-e2e	6.260M	285.5	507.0	1487.8	28.33	49.90	90.85
UPRNet-ours-plugin	0.009M	162.0	237.6	872.7	29.20	50.72	92.60

Table 3. Ablation Study on end-to-end and plug-in adapter adaptation. Models boosted by our proposed plug-in adapter require minimal finetuning parameters for adaptation, resulting in a 2 times improvement in efficiency while maintaining comparable inference efficiency and performance.

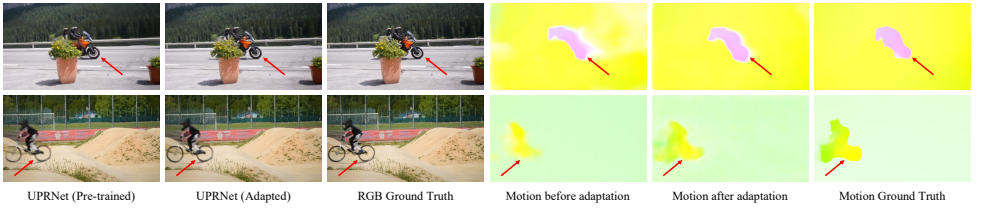


Figure 4. Motion field visualization. The VFI model boosted by our proposed motion adaptation can estimate more precise motion fields, thereby producing synthesized frames with higher quality.

Motion Field Visualization. As stated in Sec. 3.2, our proposed cycle-consistency adaptation strategy enables VFI models to fully use inter-frame consistency, and thus acquire motion patterns that are more suitable for testing scenarios. To qualitatively verify this idea, we visualize the motion fields estimated by the VFI model before and after adaptation. Specifically, we compute the optical flow between the target frame and the reference frame by RAFT [45] as motion ground truth, and compare the motion fields estimated by UPRNet [15] before and after adaptation. As shown in Figure 4, the model boosted by our proposed motion adaptation can output **smoother motion fields** and **more precise motion edges**, leading to steady improvement in the quality of synthesized images.

6 Conclusion

In this paper, we present optimisation-based video frame interpolation to tackle the generalisation challenge of VFI models and boost their performance at inference time. To this end, a test-time motion adaptation strategy that is suitable for video frame interpolation has been introduced, namely *cycle-consistency adaptation*. In order to address the efficiency drawback of motion adaptation, we further propose a lightweight yet effective plug-in *adapter* module which can be injected into the motion estimation module of existing pre-trained VFI models to refine the estimated motion flow, thus synthesizing higher-quality intermediate frames. Extensive experiments on various models and benchmarks have demonstrated the effectiveness of the proposed cycle-consistency adaptation strategy on VFI task and confirmed that the proposed plug-in adapter module can efficiently and steadily boost the performance of VFI models, even outperforming approaches with extra inputs.

Acknowledgments

This work is supported by National Natural Science Foundation of China (62271308), STCSM (22511105700, 22DZ2229005), 111 plan (BP0719010), and State Key Laboratory of UHD Video and Audio Production and Presentation.

References

[1] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3703–3712, 2019.

[2] Wenbo Bao, Wei-Sheng Lai, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(3):933–948, 2019.

[3] 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.

[4] Zhixiang Chi, Yang Wang, Yuanhao Yu, and Jin Tang. Test-time fast adaptation for dynamic scene deblurring via meta-auxiliary learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9137–9146, 2021.

[5] Myungsub Choi, Janghoon Choi, Sungyong Baik, Tae Hyun Kim, and Kyoung Mu Lee. Scene-adaptive video frame interpolation via meta-learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9444–9453, 2020.

[6] 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.

[7] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1801–1810, 2019.

[8] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5515–5524, 2016.

[9] Ping Hu, Simon Niklaus, Stan Sclaroff, and Kate Saenko. Many-to-many splatting for efficient video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3553–3562, 2022.

[10] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2015.

[11] 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.

[12] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8981–8989, 2018.

[13] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2462–2470, 2017.

[14] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9000–9008, 2018.

[15] 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.

[16] Xin Jin, Longhai Wu, Guotao Shen, Youxin Chen, Jie Chen, Jayoon Koo, and Cheul-hee Hahm. Enhanced bi-directional motion estimation for video frame interpolation. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pages 5049–5057, 2023.

[17] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (TOG)*, 35(6): 1–10, 2016.

[18] Tarun Kalluri, Deepak Pathak, Manmohan Chandraker, and Du Tran. Flavr: Flow-agnostic video representations for fast frame interpolation. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pages 2071–2082, 2023.

[19] 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.

[20] Zihang Lai and Weidi Xie. Self-supervised video representation learning for correspondence flow. In *Proceedings of the British Machine Vision Conference*, 2019.

[21] Hyeongmin Lee, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. Adacof: Adaptive collaboration of flows for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5316–5325, 2020.

[22] Yizhuo Li, Miao Hao, Zonglin Di, Nitesh Bharadwaj Gundavarapu, and Xiaolong Wang. Test-time personalization with a transformer for human pose estimation. *Advances in Neural Information Processing Systems*, 34:2583–2597, 2021.

[23] Yihao Liu, Liangbin Xie, Li Siyao, Wenxiu Sun, Yu Qiao, and Chao Dong. Enhanced quadratic video interpolation. In *Proceedings of the European Conference on Computer Vision Workshops*, pages 41–56, 2020.

[24] Yu-Lun Liu, Yi-Tung Liao, Yen-Yu Lin, and Yung-Yu Chuang. Deep video frame interpolation using cyclic frame generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8794–8802, 2019.

[25] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *Proceedings of the International Conference on Computer Vision*, pages 4463–4471, 2017.

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations*, 2019.

[27] Liying Lu, Ruizheng Wu, Huaijia Lin, Jiangbo Lu, and Jiaya Jia. Video frame interpolation with transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3532–3542, 2022.

[28] Tomer Michaeli and Michal Irani. Nonparametric blind super-resolution. In *Proceedings of the International Conference on Computer Vision*, pages 945–952, 2013.

[29] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1710, 2018.

[30] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5437–5446, 2020.

[31] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 670–679, 2017.

[32] 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.

[33] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. In *Proceedings of the European Conference on Computer Vision*, pages 109–125, 2020.

[34] Junheum Park, Chul Lee, and Chang-Su Kim. Asymmetric bilateral motion estimation for video frame interpolation. In *Proceedings of the International Conference on Computer Vision*, pages 14539–14548, 2021.

[35] Tomer Peleg, Pablo Szekely, Doron Sabo, and Omry Sendik. Im-net for high resolution video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2398–2407, 2019.

[36] 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.

[37] Fitsum Reda, Janne Kontkanen, Eric Tabellion, Deqing Sun, Caroline Pantofaru, and Brian Curless. Film: Frame interpolation for large motion. In *Proceedings of the European Conference on Computer Vision*, 2022.

[38] Fitsum A Reda, Deqing Sun, Aysegul Dundar, Mohammad Shoeybi, Guilin Liu, Kevin J Shih, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Unsupervised video interpolation using cycle consistency. In *Proceedings of the International Conference on Computer Vision*, pages 892–900, 2019.

[39] Zhihao Shi, Xiaohong Liu, Kangdi Shi, Linhui Dai, and Jun Chen. Video frame interpolation via generalized deformable convolution. *IEEE Transactions on Multimedia*, 24:426–439, 2021.

[40] Zhihao Shi, Xiangyu Xu, Xiaohong Liu, Jun Chen, and Ming-Hsuan Yang. Video frame interpolation transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 17482–17491, 2022.

[41] Assaf Shocher, Nadav Cohen, and Michal Irani. "zero-shot" super-resolution using deep internal learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3118–3126, 2018.

[42] Hyeonjun Sim, Jihyong Oh, and Munchurl Kim. Xvfi: extreme video frame interpolation. In *Proceedings of the International Conference on Computer Vision*, pages 14489–14498, 2021.

[43] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018.

[44] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *Proceedings of the International Conference on Machine Learning*, pages 9229–9248, 2020.

[45] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision*, pages 402–419, 2020.

[46] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2566–2576, 2019.

[47] Chao-Yuan Wu, Nayan Singhal, and Philipp Krahenbuhl. Video compression through image interpolation. In *Proceedings of the European Conference on Computer Vision*, pages 416–431, 2018.

[48] Haiping Wu and Xiaolong Wang. Contrastive learning of image representations with cross-video cycle-consistency. In *Proceedings of the International Conference on Computer Vision*, pages 10149–10159, 2021.

[49] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. *Advances in Neural Information Processing Systems*, 32, 2019.

[50] 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.

[51] Guozhen Zhang, Yuhan Zhu, Hongya Wang, Youxin Chen, Gangshan Wu, and Limin Wang. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023.

[52] Qinye Zhou, Ziyi Li, Weidi Xie, Xiaoyun Zhang, Yan-Feng Wang, and Ya Zhang. A simple plugin for transforming images to arbitrary scales. In *Proceedings of the British Machine Vision Conference*, 2022.

[53] Tinghui Zhou, Philipp Krahenbuhl, Mathieu Aubry, Qixing Huang, and Alexei A Efros. Learning dense correspondence via 3d-guided cycle consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 117–126, 2016.

[54] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the International Conference on Computer Vision*, pages 2223–2232, 2017.