TTVFI: Learning Trajectory-Aware Transformer for Video Frame Interpolation

Chengxu Liu, Huan Yang, Jianlong Fu, and Xueming Qian, *Member, IEEE*

***Abstract*— Video frame interpolation (VFI) aims to synthesize an intermediate frame between two consecutive frames. State-of-the-art approaches usually adopt a two-step solution, which includes 1) generating locally-warped pixels by calculating the optical flow based on pre-defined motion patterns (e.g., uniform motion, symmetric motion), 2) blending the warped pixels to form a full frame through deep neural synthesis networks. However, for various complicated motions (e.g., non-uniform motion, turn around), such improper assumptions about pre-defined motion patterns introduce the inconsistent warping from the two consecutive frames. This leads to the warped features for new frames are usually not aligned, yielding distortion and blur, especially when large and complex motions occur. To solve this issue, in this paper we propose a novel Trajectory-aware Transformer for Video Frame Interpolation (TTVFI). In particular, we formulate the warped features with inconsistent motions as query tokens, and formulate relevant regions in a motion trajectory from two original consecutive frames into keys and values. Self-attention is learned on relevant tokens along the trajectory to blend the pristine features into intermediate frames through end-to-end training. Experimental results demonstrate that our method outperforms other state-of-the-art methods in four widely-used VFI benchmarks. Both code and pre-trained models will be released at https://github.com/ChengxuLiu/TTVFI.**

***Index Terms*— Video frame interpolation, trajectory-aware transformer, consistent motion field.**

I. Introduction

VIDEO frame interpolation (VFI) aims to synthesize non-existent frames between two consecutive frames. It is a fundamental problem in computer vision and can be applied to numerous applications, including slow-motion video generation [18], frame rate upconversion [3], video compression [49], and view synthesis [11]. From a methodology perspective, unlike other image/video restoration tasks that usually recover enhanced images/videos from low-quality visual information on spatial dimensions, VFI tasks pay more attention to exploiting temporal motion information and synthesizing high-quality texture details in interpolated frames. As shown in Fig. 1, if detailed textures to recover the target frame can be discovered and leveraged at adjacent frames, video qualities can be greatly enhanced.

Recently, classical frame interpolation algorithms synthesize the interpolated results either by predicting the blending kernels [6], [23], [33], [34] or with help of motion estimation networks [1], [16], [31], [32], [36], [37]. The former makes attempts to predict the blending kernels, and the interpolated result is obtained by filtering operation. However, the kernel size directly restricts the motion that the model can capture. Capturing larger motions with larger kernel size (e.g., 51×51 in [34]) results in heavy memory and computation cost. For the latter, benefiting from significant progress of motion estimation [29], [46], the typical frame interpolation algorithms use auxiliary of optical flow to synthesize the interpolated results, such as DAIN [1], BMBC [36], and ABME [37]. Nevertheless, the accuracy of the motion field and the manner of intermediate frame synthesis remain the great challenges that limit the effectiveness of VFI.

In particular, to solve this challenge, recent years have witnessed an increasing number of advanced algorithms [1], [31], [32], [36], [37] to investigate the effects of motion field (i.e., optical flow) in video frame interpolation. Typical algorithms [36], [37], [50] assume some pre-defined motion patterns (e.g., uniform motion, symmetric motion, asymmetric motion) to estimate optical flow and feed the warped frames obtained via bi-directional flow-based warping into a synthesis network of intermediate frames. However, there are still some problems as follows: 1) In estimating optical flow, for some challenging scenes and regions (e.g., fast-moving, non-uniform motion, turn around), the pre-defined single variety of motion patterns cannot handle multiple kinds of complex motions and may produce inaccurate or inconsistent motion fields (indicated in Fig. 7(a) and (b)). 2) Existing synthesis networks use the warped frame as input focusing on achieving overall interpolation averaged over all regions of intermediate results. However, the flow-based warping will produce inaccurate texture synthesis in the regions with inconsistent motion fields (indicated in Fig. 1), which is common and necessary in VFI. Therefore, such designs lack a necessary design to improve the interpolation results in important regions and may produce

Manuscript received 6 February 2023; revised 13 June 2023; accepted 30 July 2023. Date of publication 11 August 2023; date of current version 22 August 2023. This work was supported in part by NSFC under Grant 62272380 and Grant 62103317; in part by the Science and Technology Program of Xi'an, China, under Grant 21RGZN0017; and in part by Microsoft Research. The work of Chengxu Liu was supported by Microsoft Research Asia. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Zhu Li. *(Corresponding authors: Huan Yang; Xueming Qian.)*

Chengxu Liu is with the School of Information and Communication Engineering, Xi'an Jiaotong University, Xi'an 710049, China, and also with Shaanxi Yulan Jiuzhou Intelligent Optoelectronic Technology Company Ltd., Xi'an 710000, China (e-mail: liuchx97@gmail.com).

Huan Yang and Jianlong Fu are with Microsoft Research Asia, Beijing 100080, China (e-mail: huayan@microsoft.com; jianf@microsoft.com).

Xueming Qian is with the Ministry of Education Key Laboratory for Intelligent Networks and Network Security, School of Information and Communication Engineering, and the SMILES Laboratory, Xi'an Jiaotong University, Xi'an 710049, China, and also with Shaanxi Yulan Jiuzhou Intelligent Optoelectronic Technology Company Ltd., Xi'an 710000, China (e-mail: qianxm@mail.xjtu.edu.cn).

Digital Object Identifier 10.1109/TIP.2023.3302990

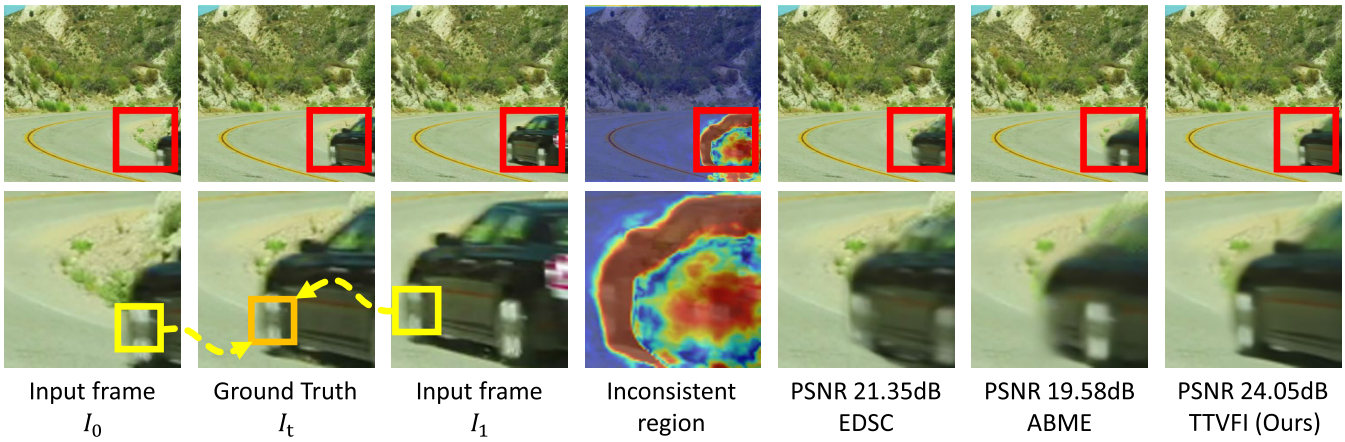


Fig. 1. A comparison between TTVFI and other SOTA methods: EDSC [6] and ABME [37]. Zoom in to see better visualization (indicated by red). TTVFI focuses on the inconsistent motion regions (indicated by warmer color), attention is learned on relevant tokens along the trajectory (indicated by yellow) to blend the pristine features into intermediate frames (indicated by orange).

distortion and blurring (e.g., the ABME [37] in Fig. 1). A more promising solution is to explore a proper synthesis network for generating intermediate results by introducing pristine features of the original input frames.

Besides, inspired by the recent significant progress of Transformer [47] in video restoration [4], [24], [42], [53], VSR-Transformer [4] and TTVSR [24] propose to use Transformer to generate the enhanced and high-resolution object in recovered video. In VFI, VFIT [42] and VFIformer [28] propose to use Transformer to extract deep hierarchical features, and predict the blending kernels for interpolating results. However, they learn various motion patterns between frames directly through the attention mechanism, which benefits from the long-range dependent learning capability of the Transformer itself and has not exploited the potential of the attention mechanism in object modeling and improving interpolation results. Therefore, in VFI, utilizing Transformer to synthesize high-quality texture details and pleasing interpolation results remains a great challenge.

In this paper, we propose a novel Trajectory-aware Transformer to achieve more accurate and effective feature learning in Video Frame Interpolation (TTVFI), as shown in Fig. 2. The key insight is to focus on the regions with inconsistent motion (e.g., the inconsistent region in Fig. 1), and allow features to be learned from the original input frames through the attention mechanism. In particular, we propose a consistent motion learning component in trajectory-aware Transformer at first, as shown in Fig. 3, to obtain motion fields, which can be used to generate a group of inter-frame motion trajectories. Then, the trajectories and motion fields are used to formulate the two kinds of visual tokens. They come from the original input frames and warped frames and learn on the relevance of them in regions with inconsistent and consistent motion, respectively. Finally, once the tokens have been obtained, TTVFI learns relevant features by calculating self-attention in regions with inconsistent and consistent motion. The output of TTVFI can be stacked in multi-scale to further boost feature representation of intermediate results.

Compare with VFIformer [28] and VFIT [42] that use Transformer to learn various motion patterns between frames directly through the attention mechanism and predict the blending kernels for interpolating results. Our proposed TTVFI models the motion as a set of per-defined inter-frame trajectories, selects features from the input frames along the trajectory and synthesize richer textures in a trajectory-based way. This manner exploits the potential of feature restoration in the synthesis network and improves interpolation results through well-designed visual tokens along the motion trajectory. Compared with TTVSR [24] that proposes long-range motion trajectories to effective utilization of information from more distant frames in video super-resolution. We propose inter-frame sparse trajectories to indicate position changes in the space of different regions during frame interpolation, which allows pristine features from the input frames to be found during interpolation of different regions.

Our contributions are summarized as follows:

- We propose a novel trajectory-aware Transformer, which enables more accurate features learning of synthesis network by introducing Transformer into VFI tasks. Our method focuses on regions of video frames with motion consistency differences and performs attention with two kinds of well-designed visual tokens along the motion trajectory.
- We propose a consistent motion learning module to generate the consistent motion in trajectory-aware Transformer, which is used to generate the trajectories and guide the learning of the attention mechanism in different regions.
- Extensive experiments demonstrate that the proposed TTVFI can outperform existing state-of-the-art methods in four widely-used VFI benchmarks.

The rest of the paper is organized as follows. Related work is reviewed in Sec. II. The proposed trajectory-aware Transformer is elaborated in Sec. III. Experimental evaluation, analysis, and ablation study are presented in Sec. IV. The discussion of the related parameters and component are presented in Sec. V. The limitations and failure cases are elaborated in Sec. VI. Finally, we conclude this work in Sec. VII.

II. Related Work

In this section, we mainly introduce the related work on video frame interpolation (VFI). Additionally then, we give

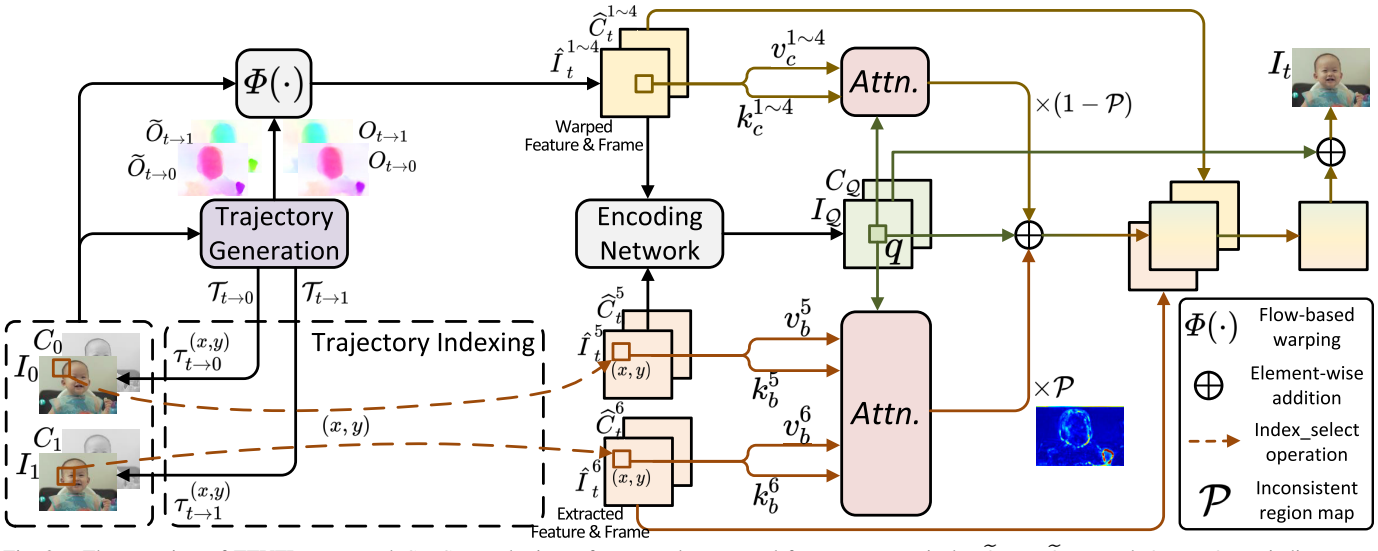


Fig. 2. The overview of TTVFI. I_0, I_1 and C_0, C_1 are the input frames and contextual features, respectively. $\widetilde{O}_{t\to 0}$, $\widetilde{O}_{t\to 1}$ and $O_{t\to 0}$, $O_{t\to 1}$ indicate two kinds of motion fields. $\tau_{t\to 0}^{(x,y)}$ and $\tau_{t\to 1}^{(x,y)}$ are elements of the trajectories set $\mathcal{T}_{t\to 0}$ and $\mathcal{T}_{t\to 1}$ at the start point (x, y), respectively. q, k, and v indicate the query, key, and value, respectively. $\mathcal{P}$ indicates the inconsistent region map. I_t indicates the output intermediate frame.

a brief overview of blurry VFI, quality assessment of VFI, visual Transformer and their application in various fields.

A. Video Frame Interpolation

Video frame interpolation is a classical problem in various image processing and computer vision tasks [11], [18], [41], [49], [56]. In this section, we focus on recent VFI algorithms, which can be classified into two paradigms: kernel-based [6], [23], [33], [34] methods and flow-based [1], [2], [13], [16], [25], [27], [31], [32], [36], [37], [51] methods.

1) Kernel-Based Video Interpolation: The kernel-based methods make attempts to estimate the blending kernels using CNNs [33], [34] or deformable convolutions [5], [6], [8], and the interpolated result is obtained by filtering operation. Typically, AdaConv [34] and SepConv [33] predict spatially-adaptive and separable interpolation kernels respectively to aggregate each pixel from the neighborhood. DSepConv [5] and EDSC [6] propose adaptively estimate kernels using deformable separable convolution to extend the receptive field of the pre-defined kernel and focusing on more relevant pixels. To solve the degrees of freedom limitations in complex motions, AdaCoF [23] propose to estimates both kernel weights and offset vectors for each pixel.

However, the kernel size directly restricts the motion that the model can capture. Capturing larger motions with larger kernel size (e.g., 51×51 in [34]) results in heavy memory and computation cost.

2) Flow-Based Video Interpolation: Unlike relying on kernel estimation, the flow-based methods have been developed most actively and usually consist of two steps: 1) warping the input frames based on the optical flow from the motion estimation network, 2) blending the warped frames through the synthesis network. The flow-based methods focus on generating more accurate motion to warp the input frames, and contain two algorithms using forward warping [32] and backward warping [1], [13], [16], [31], [36], [37]. Typically, SoftSplat [32] proposes softmax splatting to address the conflict of mapping multiple pixels to the same target location in forward warping, but suffers from holes pixels. For the methods using backward warping, CtxSyn [31] presents a context-aware synthesis approach to effectively blend the two warped frames. DAIN [1] introduces the depth information to deal with the holes or overlay caused by occlusion. FeatureFlow [13] proposes to predict the optical flow of features to handle the interpolation of complex dynamic scenes. Further, to estimate the motion more accurately, BMBC [36] and ABME [37] pre-define symmetric and asymmetric bilateral motion patterns to estimate the optical flow between video frames. All these methods reconstruct intermediate frame by blending the warped frames through the synthesis network.

However, for some challenging scenes, the pre-defined motion patterns may produce inaccurate or inconsistent motion fields, resulting in distortion and blurring. Besides, these methods focus on achieving overall interpolation averaged over all regions of the intermediate result by the synthesis network and lack a necessary design for improving the interpolation result in important regions.

B. Blurry Video Frame Interpolation

Besides the flow-/kernel-based classes, recently, the joint problem of deblurring and interpolation has also received increasing attention [19], [35], [40], [41], [57], [58]. TNTT [19] takes an approximate recurrent approach and first extracts several clear keyframes which are then subsequently used to generate intermediate sharp frames. BIN [40] and its extended version PRF [41] adopts a recurrent pyramid structure to propagate the temporal information over time and obtain both the deblurred input and the center-frame. DeMFI [35] proposes flow-guided attentive-correlation-based feature bolstering module and recursive boosting techniques to convert blurred videos to sharp videos. Recently, BiT [57] introduces dual-end temporal supervision and temporally symmetric ensembling strategies and proposes a blur interpolation Transformer to effectively unravel the underlying temporal

correlation encoded in blur. Although our method is not specially designed for blurred input images, it still exhibits strong robustness in addition.

C. Quality Assessment for VFI

Quality assessment is essential in vision tasks due to the presence of quality changes in visual image/video processing tasks [30], [54]. In VFI, besides PSNR and SSIM, which are most commonly used to measure interpolation performance, various methods for quality assessment have emerged in recent years [9], [15]. VFIPS [15] learns perceptual features directly from videos and present a dedicated perceptual quality metric for measuring interpolated result. FloLPIPS [9] captures the perceptual degradation in extracted image feature space and presents a bespoke full reference video quality metric for VFI. However, the existing models use non-uniform databases and are still far from developing a standard metric.

D. Visual Transformer

Recently, due to its advanced learning capabilities, Transfomer [47] as a new attention-based paradigm for modeling relationships between visual tokens in many computer vision tasks, such as image classification [10], [26], super-resolution [24], [52] and so on. In VFI tasks, benefiting from the long-range dependence learning capability of the Transformer, VFIT [42] and VFIformer [28] predict the blending kernels for achieving interpolation. Notably, totally different from them to learn long-range motion through attention mechanism directly, our proposed trajectory-aware Transformer first models the motion as a set of per-defined inter-frame trajectories, and then blends the pristine features into the intermediate frames by self-attention along the trajectories. In general, Transformer can be well-used for visual object recovery in the tasks of video reconstruction.

Besides, Motionformer [38] and TTVSR [24] also propose the Transformer-based structure combining trajectory information for video action recognition and video super-resolution. However, it is worth noting that the implications of trajectories and the methods proposed in these works are completely different from ours. Motionformer [38] proposes to utilize implicitly determined trajectories in the attention mechanism for better aggregation of video information. TTVSR [24] proposes long-range motion trajectories in video to enable effective utilization of information from more distant frames in video super-resolution tasks. In this paper, we propose inter-frame sparse trajectories to indicate position changes in the space of different regions during frame interpolation. Based on this formulation, we propose a novel trajectory-aware Transformer that improves interpolated results by extracting pristine features from the input frame and performing attention in different regions along the trajectories.

III. Trajectory-Aware Transformer

A. Overview

Existing works [1], [31], [37] warp the input frames by the optical flow with pre-defined motion patterns, and lack a necessary design for the important synthesis network. Therefore, we propose the trajectory-aware Transformer to mitigate the distortion and blur caused by inconsistent warping and synthesize the interpolation results.

As shown in Fig. 2, TTVFI takes two successive frames I_0, I_1 and extracted context feature C_0, C_1 as input, and generates an intermediate frame $I_t, t \in (0, 1)$. Specifically, we first propose a trajectory generation module to obtain the motion field $\mathcal{O}$ and trajectory $\mathcal{T}$ between two successive video frames. Then, we use the flow-based warping $\Phi(\cdot)$ and trajectory indexing $Idx(\cdot)$ to generate two different features from different sources separately, and formulate them into two kinds of visual tokens by encoding network, named as consistent tokens v_c, k_c and boundary tokens v_b, k_b. Finally, we perform trajectory-aware attention $A_{traj}(\cdot)$ in regions with different motion consistency (indicated by $\mathcal{P}$). The attention results are fed into a feed-forward network $FFN(\cdot)$ consisting of residual blocks (omitted for brevity in Fig. 2), and output the feature of the intermediate frame I_t. In summary, the trajectory-aware Transformer $T_{traj}(\cdot)$ can be formulated as:

$$\begin{aligned} &\mathrm{T}_{traj}(\mathcal{Q}, \mathcal{K}, \mathcal{V}) \\ &\quad = \mathrm{FFN}(\mathrm{A}_{traj}((\mathcal{Q}, \mathcal{K}_c, \mathcal{V}_c), (\mathcal{Q}, \mathcal{K}_b, \mathcal{V}_b), \mathcal{T}) + \mathcal{Q}), \end{aligned} \tag{1}$$

where $(\mathcal{K}_c, \mathcal{V}_c)$ and $(\mathcal{K}_b, \mathcal{V}_b)$ indicate the consistent tokens set and boundary tokens set, respectively. $\mathcal{T}$ is the motion trajectory. $\mathcal{Q}$, $\mathcal{K}$, $\mathcal{V}$ indicate the generic element queries, keys and values entered into Transformer. Note that we stack trajectory-aware Transformer on multiple scales to facilitate the learning of features. Here, we describe this structure only at one scale for brevity.

B. Trajectory Generation

To alleviate the effects of inconsistent warping. We first estimate the consistent motion with a proposed consistent motion learning component. Then the consistent motion is further used to generate the motion trajectories of tokens.

1) Consistent Motion Learning Component: In video frame interpolation, since the intermediate frame I_t is not available, it is not possible to directly obtain the motion field between the input frames I_0, I_1 and the intermediate frame I_t.

Existing methods [36], [37] estimate an approximated motion by pre-defining some specific motion patterns (i.e., symmetric bilateral motion and asymmetric bilateral motion), which can be represented as:

$$\begin{aligned} O_{t\to 0} &= -t(1-t)\cdot O_{0\to 1} + t^2 \cdot O_{1\to 0}, \\ O_{t\to 1} &= (1-t)^2 \cdot O_{0\to 1} - t(1-t)\cdot O_{1\to 0}, \end{aligned} \tag{2}$$

where $O_{t\to 0}$ and $O_{t\to 1}$ are approximated by combining $O_{0\to 1}$ and $O_{1\to 0}$, which indicate the motion field between I_0 and I_1. For fair comparison, we follow previous works [1], [36], [37] to obtain $O_{0\to 1}$ and $O_{1\to 0}$ by PWC-Net [46]. However, the approximated motion ignores the consistency between two consecutive frames and leads to incorrect results for challenging scenes.

Therefore, as shown in Fig. 3, we propose a consistent motion learning component based on the approximated

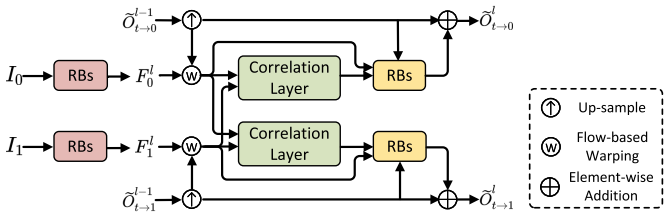


Fig. 3. The architecture of the consistency motion learning component.

motion, which is integrated into the two largest scales of the PWC-Net [46]. The initial input of the component comes from the Eq. (2). The output of the component in the last scale is the consistent motion $\widetilde{O}_{t\to 0}$ and $\widetilde{O}_{t\to 1}$ with opposite directions simultaneously. Specifically, $\widetilde{O}^{l-1}_{t\to 0}$ and $\widetilde{O}^{l-1}_{t\to 1}$ indicate the motion from level $l-1$, it is up-sampled to warp the features F^l_0 and F^l_1 from level l. Among them, for ensuring the efficiency and local smoothness of the optical flow in the upsampling process, we follow the existing methods [36], [37] and use a bilinear upsampling of the optical flow. The matching costs of the two warped features are then computed in the correlation layer [46] (indicated by green) in an interactive way. Then, for getting $\widetilde{O}^l_{t\to 0}$, we use the output cost volume from correlation layer, the warped feature from F^l_0 and the up-sampled motion from $\widetilde{O}^{l-1}_{t\to 0}$ as input to generate the residual field. Finally, the residual field is added to the up-sampled $\widetilde{O}^{l-1}_{t\to 0}$ to yield the $\widetilde{O}^l_{t\to 0}$. $\widetilde{O}^l_{t\to 1}$ can be obtained in the same way. The stacked residual block (indicated by red and yellow) is the same as the residual block used in PWC-Net.

The core advantage of this component is that the two input optical flow in opposite directions can be optimized with each other and output simultaneously. Compared with approximated motion, the consistent motion has better temporal coherence, which helps in better trajectory generation in the following part.

2) Trajectory Formulation: The trajectories $\mathcal{T}_{s\to e}$ in our approach can be formulated as a set of trajectories, in which each trajectory $\tau^{(x,y)}_{s\to e}$ contains two coordinates. The start point is associated with the coordinate of the token at position (x, y) at time s and the endpoint is associated with the coordinate of the token at time e. They can be defined as:

$$\begin{aligned}\mathcal{T}_{s\to e} &= \{\tau^{(x,y)}_{s\to e} \,|\, x \in \{1,\ldots,H\}, y \in \{1,\ldots,W\}\},\\ \tau^{(x,y)}_{s\to e} &= \langle (x, y), (x_{s\to e}, y_{s\to e})\rangle,\end{aligned} \tag{3}$$

where $(x_{s\to e}, y_{s\to e})$ represents the coordinate transformation of the token at position (x, y) from time s to e. H and W represent the height and width of the features, respectively. Specifically, the trajectories $\mathcal{T}_{s\to e}$ can be calculated by:

$$\mathcal{T}_{s\to e} = \gamma(M + \widetilde{O}_{s\to e}), \tag{4}$$

where M represents a two-dimensional meshgrid matrix[1] of the same size as consistent motion $\widetilde{O}_{s\to e}$. $\gamma(\cdot)$ indicates the rounding operation to align the coordinates of the tokens at the endpoint of the trajectories. The difference between trajectories and motion fields is how they affect the feature calculation. Motion fields generate features by pixel-level interpolation during warping. In contrast, trajectories describe the sequence of changes in coordinates between relevant tokens, allowing the extraction of patch-level pristine features and avoiding textures damaged by warping.

C. Token Generation

To build visual tokens from different sources separately, we first generate two different features, named as warped features and extracted features. Then they are used to build query, key, and value tokens respectively.

Specifically, the warped features $\widehat{C}^k_t, k \in \{1, 2, 3, 4\}$ are obtained by bi-directional flow-based warping. For regions with consistent motion, the warped features from two consecutive frames are well aligned, which can be obtained by:

$$\begin{aligned}\widehat{C}^1_t &= \Phi_b(O_{t\to 0}, C_0), \quad \widehat{C}^2_t = \Phi_b(O_{t\to 1}, C_1),\\ \widehat{C}^3_t &= \Phi_b(\widetilde{O}_{t\to 0}, C_0), \quad \widehat{C}^4_t = \Phi_b(\widetilde{O}_{t\to 1}, C_1),\end{aligned} \tag{5}$$

where $\Phi_b(\cdot)$ is the backward warping. C_0 and C_1 are the contextual features obtained from the input frames by two convolutional layers. The extracted features $\widehat{C}^k_t, k \in \{5, 6\}$ are obtained by extracting the features of input frames along the trajectories. For regions with inconsistent motion, the extracted features can introduce the pristine features from the original input, which can be obtained by:

$$\widehat{C}^5_t = \mathrm{Idx}(\mathcal{T}_{t\to 0}, C_0), \quad \widehat{C}^6_t = \mathrm{Idx}(\mathcal{T}_{t\to 1}, C_1), \tag{6}$$

where $\mathrm{Idx}(\cdot)$ denotes the operation of trajectory indexing (i.e., $index\_select$[2]). The warped frames $\widehat{I}^k_t, k \in \{1, 2, 3, 4\}$ and extracted frames $\widehat{I}^k_t, k \in \{5, 6\}$ can be obtained in the same way.

1) Query: We build queries by the output feature from a proposed encoding network. Inspired by previous work [31], [37], the encoding network can be split into a GridNet [12] to generate filters and a dynamic local convolution [17] to output feature of intermediate frame.

First, we use the GridNet to generate local blending filters by inputting all the features $\widehat{C}^k_t, k \in \{1, 2, 3, 4, 5, 6\}$ and frames $\widehat{I}^k_t, k \in \{1, 2, 3, 4, 5, 6\}$ obtained above. Then, the generated filters by the GridNet can be denoted as $H_{(x,y)}(i, j, k)$, where $(i, j, k) \in \{-1, 0, 1\} \times \{-1, 0, 1\} \times \{1, 2, 3, 4, 5, 6\}$ is the local coordinate around (x, y) in the features. The range of (i, j) is dependent on the kernel size generated by GridNet. The dynamic local convolution uses the generated filters to yield the feature of intermediate frame $C_{\mathcal{Q}}$ by:

$$C_{\mathcal{Q}} = \sum_{k=1}^{6}\sum_{i=-1}^{1}\sum_{j=-1}^{1} H_{(x,y)}(i, j, k) \cdot \widehat{C}^k_t(x + i, y + j), \tag{7}$$

where the coefficients are normalized by $\sum_k \sum_i \sum_j H_{(x,y)}(i, j, k) = 1$ to ensure the magnitude of the pixels after convolution. By introducing the information from neighboring pixels, the convolution can compensate for the inconsistent motion to an extent. The intermediate frame $I_{\mathcal{Q}}$ also can be obtained in the same way. Finally, this feature and

[1] Where the matrix index is equal to the element (i.e., $M(x, y) = (x, y)$).

[2] The $index\_select$ function implemented in PyTorch.

frame are fed into an embedding layer of one convolutional layer to build the queries. This process can be represented as:

$$\mathcal{Q} = E(\text{Concat}(C_{\mathcal{Q}}, I_{\mathcal{Q}})), \tag{8}$$

where Concat$(\cdot)$ and $E(\cdot)$ denote the concatenate operation and the embedding layer, respectively.

2) Key and Value: We formulate input frames into two kinds of visual tokens, named as consistent tokens and boundary tokens.

In particular, consistent tokens $(\mathcal{K}_c, \mathcal{V}_c)$ focus on the regions where the motion is consistent and well coherent. Thus, consistent tokens can be accurately generated by the warped features $\widehat{C}_t^k, k \in \{1, 2, 3, 4\}$ and frames $\widehat{I}_t^k, k \in \{1, 2, 3, 4\}$. This process can be represented as:

$$\mathcal{K}_c = \mathcal{V}_c = E(\text{Concat}(\widehat{C}_t^k, \widehat{I}_t^k)), \quad k \in \{1, 2, 3, 4\}. \tag{9}$$

Boundary tokens $(\mathcal{K}_b, \mathcal{V}_b)$ focus on the regions with inconsistent motion, which mainly appear at the boundaries of moving instances. The inaccurate warping caused by inconsistent motion can destroy the pristine features in the original input frames. Therefore, we use the extracted features $\widehat{C}_t^k, k \in \{5, 6\}$ and frames $\widehat{I}_t^k, k \in \{5, 6\}$ to construct the boundary tokens. This process can be represented as:

$$\mathcal{K}_b = \mathcal{V}_b = E(\text{Concat}(\widehat{C}_t^k, \widehat{I}_t^k)), \quad k \in \{5, 6\}. \tag{10}$$

Based on the two kinds of well-designed tokens, the model can perform attention mechanisms in different regions according to the motion consistency.

D. Trajectory-Aware Attention

To mitigate the distortion and blur caused by inconsistent motion, we further introduce an inconsistent region map $\mathcal{P}$ to guide the attention calculation.

1) Inconsistent Region Map: The inconsistent region map $\mathcal{P}$ is the same size as the input frame and indicates a confidence measure of motion consistency for different regions. It can be obtained as follow:

$$\mathcal{P} = 2 \cdot \text{Sigmoid}(|\widetilde{O}_{t\to 0} + \widetilde{O}_{t\to 1}|) - 1, \tag{11}$$

where Sigmoid$(\cdot)$ is the sigmoid function for normalization. The sum of the optical flows $|\widetilde{O}_{t\to 0} + \widetilde{O}_{t\to 1}|$ in opposite directions reflects the coherence of motion. For consistent regions, it can be completely offset and the value of $\mathcal{P}$ converges to 0. Conversely, for inconsistent motion, the value of $\mathcal{P}$ converges to 1. The purpose of the proposed inconsistent region map is to distinguish the inconsistent regions and guide the calculation of attention. More discussion of inconsistent region map can be found in the supplementary materials.

2) Attention Calculation: The input of attention mechanism consists of queries $\mathcal{Q}$, consistent tokens $(\mathcal{K}_c, \mathcal{V}_c)$, boundary tokens $(\mathcal{K}_b, \mathcal{V}_b)$, and inconsistent region map $\mathcal{P}$. In the calculation process, we first compute the dot products of the query with all keys, divide each by scaling factor and apply a softmax function to obtain the weights on the values. Then, the output of the attention mechanism can be obtained by weighting the sum of two kinds of tokens with the obtained weights and $\mathcal{P}$. We compute the features of outputs as follow:

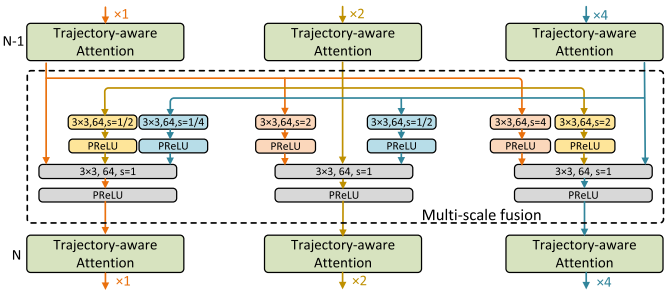


Fig. 4. The architecture of the multi-scale fusion.

$$\begin{aligned}&\text{A}_{joint}((\mathcal{Q}, \mathcal{K}_c, \mathcal{V}_c), (\mathcal{Q}, \mathcal{K}_b, \mathcal{V}_b), \mathcal{P}) \\ &\quad = (1 - \mathcal{P}) \cdot S(\frac{\mathcal{Q}\mathcal{K}_c^T}{\sqrt{d_{k_c}}})\mathcal{V}_c + \mathcal{P} \cdot S(\frac{\mathcal{Q}\mathcal{K}_b^T}{\sqrt{d_{k_b}}})\mathcal{V}_b,\end{aligned} \tag{12}$$

where $S(\cdot)$ denotes the softmax function. d_{k_c} and d_{k_b} denote the dimension of two kinds of keys. Besides, the tokens only produce a local position offset after the motion, so it is unnecessary and unrewarding in performing attention globally. Inspired by Swin Transformer [26], we perform the attention mechanism inside each shifted window to reduce the computational cost. In each attention layer, we add the feed-forward network that consists of a convolutional layer of size 3×3 and a PReLU [14] activation following it. The feed-forward network is applied to each position and considers the neighboring pixels to restructure the output feature of the trajectory-aware Transformer.

3) Multi-Scale Fusion: In the previous works [26], [52], stacking transformers in multi-layer and multi-scale has been proven to be effective. Therefore, to boost the generated feature representation of intermediate results, we stack the proposed trajectory-aware Transformer in multi-scale (i.e. $\times 1$, $\times 2$, and $\times 4$) to achieve a more powerful feature representation. Specifically, as shown in Fig. 4, to facilitate the interaction of multi-scale features, we use a multi-scale fusion module (indicated by red) in the hierarchical structure. This design enables information at each scale to exchange with each other and fuse together in a cross-scale manner. The final fused feature generates the residual that is added to the intermediate frame $I_{\mathcal{Q}}$ obtained-above to output the final intermediate frame I_t.

In general, we exploit the potential of feature restoration in the synthesis network, which is neglected in video frame interpolation. By introducing the trajectory-aware Transformer, we perform the attention mechanism along the motion trajectory with well-designed visual tokens for inconsistent regions and enable the synthetic network to learn more accurate features.

E. Training

For fair comparisons, we follow existing works [13], [36], [37] to adopt a two-stage strategy to optimize our model. In stage one, we train the consistent motion learning component to obtain the motion fields. Then, we end-to-end train the whole model in stage two.

1) Stage One: To improve the consistency of motion between the consecutive frames and the robustness of optical flow to illumination changes, we define the consistent loss L_{con} and the census loss L_{cen} as follow:

$$\begin{aligned} L_{con} &= \varphi(I_t^{GT} - \widehat{I}_{0\to t}) + \varphi(I_t^{GT} - \widehat{I}_{1\to t}) \\ &\quad + \varphi(I_1 - \Phi_b(O_{1\to t}, \widehat{I}_{0\to t})) \\ &\quad + \varphi(I_0 - \Phi_b(O_{0\to t}, \widehat{I}_{1\to t})), \end{aligned} \tag{13}$$

$$\begin{aligned} L_{cen} &= \psi(I_t^{GT}, \widehat{I}_{0\to t}) + \psi(I_t^{GT}, \widehat{I}_{1\to t}) \\ &\quad + \psi(I_1, \Phi_b(O_{1\to t}, \widehat{I}_{0\to t})) \\ &\quad + \psi(I_0, \Phi_b(O_{0\to t}, \widehat{I}_{1\to t})), \end{aligned} \tag{14}$$

where $\varphi(x) = \sqrt{x^2 + \epsilon^2}$ is the Charbonnier function [22]. The parameter ϵ is set to 1×10^{-6}. $\psi(x)$ is the census function [29], [59], which is defined as the soft Hamming distance between census transformed image patches of size 7×7. $\widehat{I}_{0\to t} = \Phi_b(\widetilde{O}_{t\to 0}, I_0)$ and $\widehat{I}_{1\to t} = \Phi_b(\widetilde{O}_{t\to 1}, I_1)$ indicate the warped frames. $O_{1\to t} = (1-t) \cdot O_{1\to 0}$ and $O_{0\to t} = t \cdot O_{0\to 1}$ denote the optical flow to warp the $\widehat{I}_{0\to t}$ and $\widehat{I}_{1\to t}$, respectively.

They ensures the consistency between the consecutive frames. Finally, the total photometric loss L_{pho} of this part is expressed as:

$$L_{pho} = L_{con} + L_{cen}. \tag{15}$$

We use the Adamax optimizer [21] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and use the batch size of 4 for 20 epochs. The initial learning rate is set as 5×10^{-5} and then reduce the learning rate by a factor of 0.2 when the losses of the testing set last for 4 epochs without decreasing.

2) Stage Two: In the second stage, we define the reconstruction loss L_{rec} between the ground truth I_t^{GT} and synthesized frame I_t to train the entire model, it is defined as:

$$L_{rec} = \varphi(I_t^{GT} - I_t). \tag{16}$$

Same as stage one, we use the same optimizer and learning rate reduction strategy. The initial learning rates of the consistent motion learning component and the trajectory-aware attention are set as 5×10^{-5} and 5×10^{-4}, respectively. We jointly train the entire model for 70 epochs. We also use the same strategies for reducing the learning rate and data augmentation as in the stage one.

IV. Experiments

A. Datasets and Metrics

1) Training Dataset: For fair comparisons, we follow existing works [13], [36], [37] to adopt a widely-used **Vimeo-90K** training set [51] to train our model. It has 51,312 triplets for training, where each triplet contains 3 consecutive video frames with a resolution of 256×448 pixels. We follow previous works [31], [36], [37] to predict the middle frame and perform data augmentation by cropping 256×256 patches, flipping horizontally, flipping vertically, and reversing the temporal order of the triplet.

2) Test Datasets: We evaluate the proposed TTVFI and compare its performance with other SOTA approaches on four widely used test sets: Vimeo-90K [51], UCF101 [27], DAVIS [39], and SNU-FILM [7].

Vimeo-90K is the Vimeo-90K testing set [51] and contains 3,782 triplets of spatial resolution 256×448.

UCF101 is the constructed test set by selecting from the human action videos dataset UCF101 [44] and contains 379 triplets of spatial resolution 256×256.

DAVIS is the constructed test set by selecting from the video object segmentation dataset DAVIS [39] and contains 30 triplets of different spatial resolutions.

SNU-FILM contains a total of 1,240 triplets videos, depending on the complexity of the motion, it has four different settings-Easy, Medium, Hard, and Extreme. Each part contains 310 triplets videos with a resolution of 1280×720.

3) Evaluation Metrics: For fair comparisons, we follow previous works [1], [6], [36], [37] to use peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [48] as a widely used metric for evaluating.

B. Comparisons With State-of-the-Art Methods

We compare TTVFI with nine classical start-of-the-art methods. These methods can be summarized into three categories: CNN-based [7], kernel-based [6], [23], [33], and flow-based video interpolation [1], [2], [16], [28], [36], [37], [51]. For fair comparisons, we obtain the performance from their original paper or reproduce results by authors' officially released models.

1) Quantitative Comparison: As shown in Tab. I, the results for each algorithm on the three test sets: Vimeo-90K [51], UCF101 [44], and DAVIS [39]. Benefiting from a pure CNN structure, CAIN [7] uses less inference time, but it does not handle motion well and has poor performance. Although the kernel-based methods (e.g., AdaCoF [23], EDSC [6]) achieve better performance than CAIN [7], the kernel size directly restricts the motion that the model can capture, resulting in heavy memory and computation cost. Thanks to the progress of motion estimation, the latest flow-based methods (e.g., ABME [37], BMBC [36], and RIFE [16]) generally perform better than the kernel-based methods. However, under some challenging conditions that decrease the accuracy of optical flow, these methods only blend warped frames through the synthesis network can lead to suboptimal performance.

TTVFI introduces more pristine features of the intermediate frame from original input frames by motion trajectories. It achieves a result of 36.54dB, 35.51dB, and 28.31dB PSNR and significantly outperforms the other algorithms for all test sets by a large margin. Specifically, on the Vimeo-90K [51] and DAVIS [39] datasets, TTVFI outperforms RIFE-Large [16] by **0.35dB** and **0.22dB**, respectively. Compared with the Transformer-based video interpolation method VFIformer [28], our method achieves a higher performance with 31.1% less number of parameters. This large margin demonstrates the power of TTVFI in feature restoration. Besides, we follow previous works [1], [36], [37] to report the runtime of interpolating a frame of size 640×480 by using an RTX 2080 Ti GPU. It should be emphasized that TTVFI

TABLE I

QUANTITATIVE COMPARISON (PSNR↑ AND SSIM↑) ON THE VIMEO-90K [51], UCF101 [44] AND DAVIS [39] DATASETS. RED INDICATES THE BEST AND BLUE INDICATES THE SECOND BEST PERFORMANCE (BEST VIEW IN COLOR)

Method	Runtime (seconds)	#Param (million)	Vimeo-90K		UCF101		DAVIS	
			PSNR(dB)↑	SSIM↑	PSNR(dB)↑	SSIM↑	PSNR(dB)↑	SSIM↑
ToFlow [51]	0.43	1.1	33.73	0.9682	34.58	0.9667	25.39	0.8555
SepConv [33]	0.20	21.6	33.79	0.9702	34.78	0.9669	26.26	0.8610
CAIN [7]	0.04	42.8	34.65	0.9730	34.91	0.9690	27.21	0.8730
MEMC [2]	0.12	70.3	34.29	0.9739	34.96	0.9682	27.25	0.8914
DAIN [1]	0.13	24.0	34.70	0.9755	34.99	0.9683	27.31	0.8932
AdaCoF [23]	0.03	22.9	34.35	0.9714	35.16	0.9680	26.59	0.8707
BMBC [36]	0.77	11.0	35.06	0.9766	35.15	0.9688	26.95	0.8872
EDSC [6]	0.07	8.9	34.84	0.9750	35.13	0.9680	26.99	0.8840
ABME [37]	0.22	18.1	36.18	0.9805	35.38	0.9698	28.07	0.8984
RIFE-Large [16]	0.08	9.8	36.19	0.9805	35.41	0.9700	-	-
VFIformer [28]	0.37	24.1	36.50	0.9816	35.43	0.9700	-	-
TTVFI	0.35	16.6	36.54	0.9819	35.51	0.9713	28.31	0.9049

TABLE II

QUANTITATIVE COMPARISON (PSNR↑ AND SSIM↑) ON THE SNU-FILM [7] DATASET. RED INDICATES THE BEST AND BLUE INDICATES THE SECOND BEST PERFORMANCE (BEST VIEW IN COLOR)

Method	Easy		Medium		Hard		Extreme	
	PSNR(dB)↑	SSIM↑	PSNR(dB)↑	SSIM↑	PSNR(dB)↑	SSIM↑	PSNR(dB)↑	SSIM↑
ToFlow [51]	39.08	0.9890	34.39	0.9740	28.44	0.9180	23.39	0.8310
SepConv [33]	39.41	0.9900	34.97	0.9762	29.36	0.9253	24.31	0.8448
CAIN [7]	39.89	0.9900	35.61	0.9776	29.90	0.9292	24.78	0.8507
MEMC [2]	39.92	0.9904	35.39	0.9779	29.93	0.9323	24.91	0.8561
DAIN [1]	39.73	0.9902	35.46	0.9780	30.17	0.9335	25.09	0.8584
AdaCoF [23]	39.80	0.9900	35.05	0.9754	29.46	0.9244	24.31	0.8439
BMBC [36]	39.90	0.9902	35.31	0.9774	29.33	0.9270	23.92	0.8432
EDSC [6]	40.01	0.9900	35.37	0.9780	29.59	0.9260	24.39	0.8430
ABME [37]	39.59	0.9901	35.77	0.9789	30.58	0.9364	25.42	0.8639
RIFE-Large [16]	40.02	0.9906	35.92	0.9791	30.49	0.9364	25.24	0.8621
VFIformer [28]	40.13	0.9907	36.09	0.9799	30.67	0.9378	25.43	0.8643
TTVFI	40.22	0.9907	36.07	0.9794	30.77	0.9397	25.67	0.8743

achieves higher performance than VFIformer [28], which is the SOTA Transformer-based method, while keeping the faster Runtime and less #Params. Such superior performances mainly benefit from the use of trajectories in attention calculation.

To further verify the generalization capabilities of TTVFI, we evaluate TTVFI on SNU-FILM [7] dataset with different complexities. As shown in Tab. II, due to the well-designed tokens and the long-range modeling capability of attention mechanism, TTVFI achieves better results in three kinds of settings, which outperforms other SOTA methods between **0.10dB** to **0.24dB**. Especially, on dataset SNU-FILM Extreme with more complex motion, TTVFI exceeds VFIformer [28] by **0.24 dB**. This is because TTVFI introduces more pristine features of the intermediate frame from original input frames by motion trajectories. The performances verify that TTVFI has strong generalization capabilities under different degrees of motion. More results for perceptual metrics can be found in the supplementary material.

2) Qualitative Comparison: To further compare the visual qualities of different approaches, we show visual results generated by TTVFI and other SOTA methods on different test sets in Fig. 5. For fair comparisons, we either directly take the original interpolated results of the author-released or use author-released models to get results. It can be observed that TTVFI has a great improvement in visual quality, especially for areas with moving instances. For example, in the fifth row in Fig. 5, TTVFI can recover the complete leg in the case of extreme motion. As the analysis mentioned above, the results verify that TTVFI can mitigate the distortion and blur caused by inconsistent warping. More visual results, video demo, and limitations can be found in the supplementary material.

C. Ablation Study

In this section, we mainly conduct the ablation study on the proposed trajectory-aware Transformer and analyze the effect of the used motion field in consistent motion learning module.

1) Trajectory-Aware Transformer: Our proposed trajectory-aware Transformer includes three important parts, "CML", "TAC", and "TAB". "CML" is the consistent motion learning module to generate the trajectory, "TA" is the trajectory-aware attention. Depending on the consistent tokens and boundary tokens being used, "TA" can be divided into "TAC" and TAB". Trajectory-aware Transformer can be interpreted as "Base+CML+TA(TAC+TAB)", and we study them together in this part. Among them, we directly use queries to generate the feature of the intermediate frame as our "Base" model. The "Base+CML" denotes that we further aggregate the features warped by consistent motion. Then we add the trajectory-aware attention based on the consistent tokens and boundary tokens as our "Base+CML+TAC" and "Base+CML+TAB" model, respectively. We add

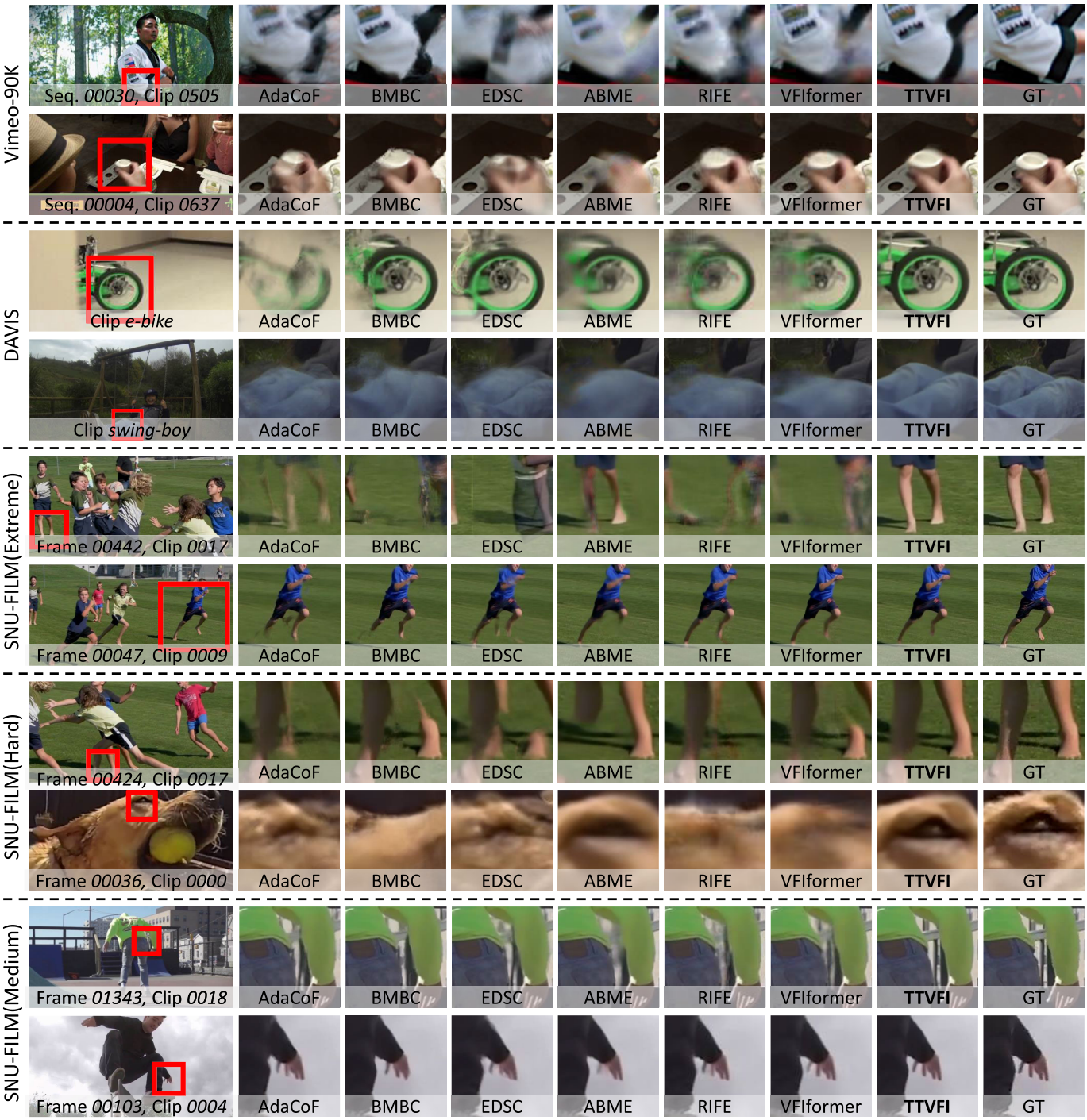


Fig. 5. Visual results on Vimeo-90K [51], DAVIS [39], and SNU-FILM [7] datasets. The frame number and method are shown at the bottom of each case. Zoom in to see better visualization.

the trajectory-aware attention based on the both tokens progressively as our "Base+CML+TAC+TAB" model. To ensure a fair comparison, we keep the parameters and FLOPs similar for each ablation by adding Residual blocks or replacing equal number of tokens.

As shown in Tab. III, the addition of CML improves the PSNR from 34.26dB to 34.95dB on Vimeo-90K [51] and from 27.28dB to 27.70dB on DAVIS [39] dataset. With the addition of TAC, the performance is improved from 34.95dB to 36.45dB on Vimeo-90K [51] and from 27.70dB to 28.21dB on DAVIS [39], respectively. With the addition of TAB, the performance is improved from 34.95dB to 36.31dB on Vimeo-90K [51] and from 27.70dB to 28.11dB on DAVIS [39], respectively. With all of them added (i.e., Base+CML+TA(TAC+TAB)), the performance has achieved 36.54dB and 28.31dB on Vimeo-90K [51] and DAVIS [39], respectively. This demonstrates the superiority of each part in TTVFI.

To further compare the visual qualities of different approaches, we further compare them as shown in Fig. 6. For fair comparisons, we use the same experimental setup for the following comparison. It can be observed that each part of

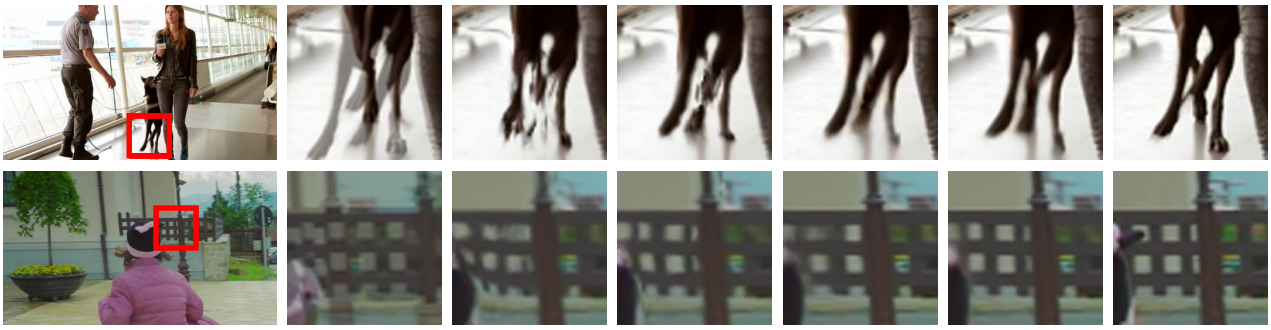


Fig. 6. Ablation study on the consistent motion learning module (CML), trajectory-aware attention with consistent tokens (TAC), and trajectory-aware attention (TA). Zoom in to see better visualization. ("TA" can be interpreted as "TAC+TAB").

Fig. 7. Visualization comparison of different motion fields. (a) symmetric bilateral motion in BMBC [36], (b) asymmetric bilateral motion in ABME [37], (c) approximated motion, (d) consistent motion. Zoom in to see better visualization.

TABLE III

ABLATION STUDY OF OUR TTVFI ON VIMEO-90K [51] AND DAVIS [39] DATASETS. CML: CONSISTENT MOTION LEARNING MODULE. TAC: TRAJECTORY-AWARE ATTENTION WITH CONSISTENT TOKENS. TAB: TRAJECTORY-AWARE ATTENTION WITH BOUNDARY TOKENS

Components				Vimeo-90K		DAVIS	
Base	CML	TAC	TAB	PSNR	SSIM	PSNR	SSIM
✓				34.26	0.9724	27.28	0.8939
✓	✓			34.95	0.9755	27.70	0.8981
✓	✓	✓		36.45	0.9815	28.21	0.9025
✓	✓		✓	36.31	0.9811	28.11	0.9003
✓	✓	✓	✓	**36.54**	**0.9819**	**28.31**	**0.9049**

TABLE IV

RESULTS OF USING DIFFERENT MOTION FIELDS ON VIMEO-90K [51] AND DAVIS [39] DATASETS

Motion field		Vimeo-90K		DAVIS	
Approximation	Consistent	PSNR	SSIM	PSNR	SSIM
✓		35.80	0.9795	27.72	0.8974
	✓	35.87	0.9799	27.91	0.8997
✓	✓	**36.54**	**0.9819**	**28.31**	**0.9049**

the TTVFI has a significant contribution to improving visual quality. For example, in the second row in Fig. 6, CML can improve the accuracy of motion fields, while TAC and TAB integrate tokens into the features to produce more complete and clearer fence structures.

2) Motion Field in Consistent Motion Learning Module: To verify the effectiveness of consistent motion generated by the consistent motion learning component. We chose different motions that described in Sec. III-B in our method to generate the token input for performing attention. It is worth noting that the unused motion field means that it is only not used for token generation, independent of whether it is obtained in the intermediate process. As shown in Tab. IV, using the consistent motion is better than approximated motion. With all of them added, the performance has achieved 36.54dB and 28.31dB on Vimeo-90K [51] and DAVIS [39], respectively. This demonstrates the superiority of consistent motion learning component in TTVFI.

To further compare the visual qualities of different motion field used, we further compare the visual differences of them as shown in Fig. 7. The consistent motion (i.e., CM) has clearer textures than other kinds of motion fields. Both quantitative and qualitative comparisons demonstrate the superiority of consistent motion.

V. DISCUSSIONS

In this section, we discuss the influence of hyper-parameters used in the attention mechanism, different motion field, multi-scale fusion structure, perceptual quality, multi-frame input, high-resolution frame input and blurry frame input.

A. The Discussions of Hyper-Parameters in Trajectory-Aware Attention

To explore the influence of hyper-parameters used in attention mechanisms that described in Sec. III-D. We discuss the different multi-head (H), window size (S), and layer number (N) in attention mechanisms, as shown in Tab. V. The impact of H is insignificant since the dimension of features is small. Proper S can effectively model spatial motion without introducing useless or insufficient information. The performance is

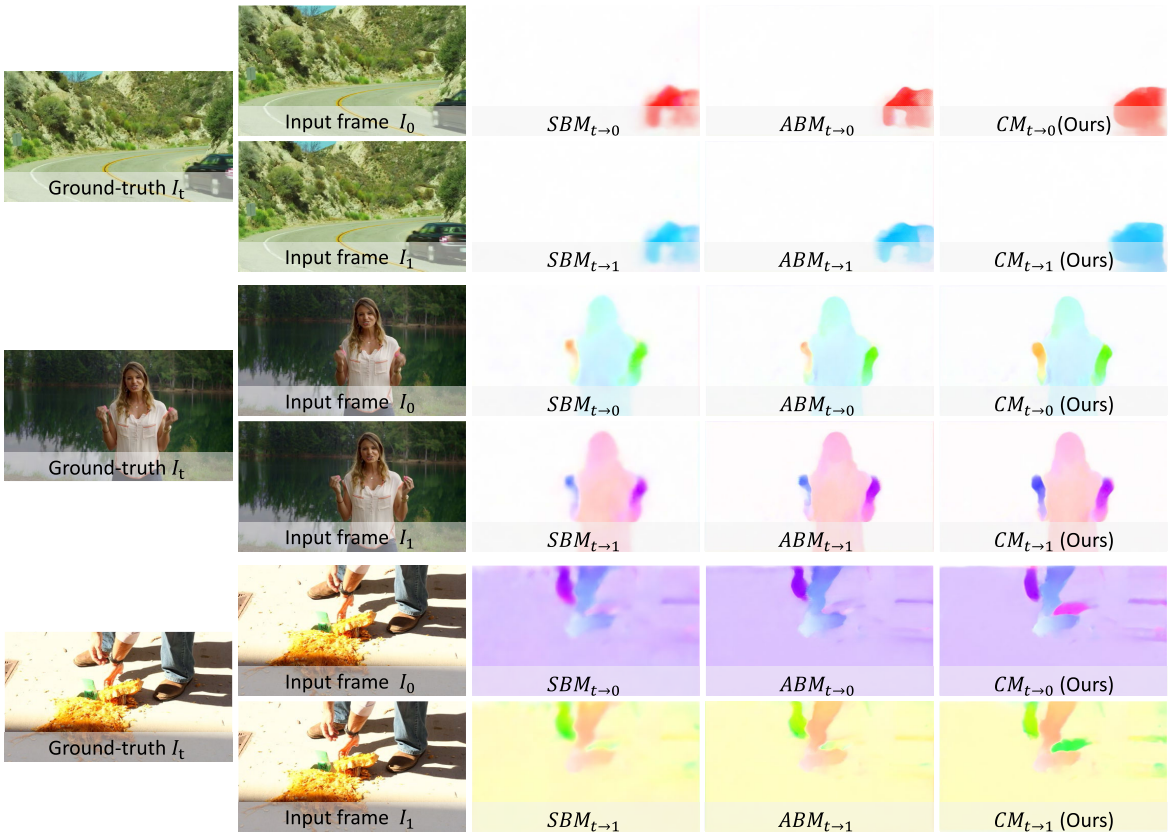


Fig. 8. Visualization comparison of consistent motion CM with symmetric bilateral motion SBM (i.e., BMBC [36]) and asymmetric bilateral motion ABM (i.e., ABME [37]).

TABLE V

RESULTS OF TRAJECTORY-AWARE ATTENTION WITH DIFFERENT MULTI-HEAD (H), WINDOW SIZE (S), AND LAYER NUMBER (N) ON VIMEO-90K [51] DATASET

H	PSNR/SSIM	S	PSNR/SSIM	N	PSNR/SSIM
2	36.52/0.9818	4	36.43/0.9814	1	36.31/0.9812
4	36.54/0.9819	8	36.54/0.9819	2	36.54/0.9819
8	36.53/0.9819	12	36.50/0.9818	3	36.58/0.9820

TABLE VI

RESULTS OF STACKED TRAJECTORY-AWARE TRANSFOMER ON MULTI-SCALES ON VIMEO-90K [51] DATASET

Scale factor ×1	Scale factor ×2	Scale factor ×4	PSNR	SSIM
✓			36.06	0.9762
✓	✓		36.31	0.9806
✓	✓	✓	36.54	0.9819

positively correlated with the N, it demonstrates the learning ability of the trajectory-aware attention. However, a deeper hierarchical structure with limited improvements will introduce heavy memory and computation cost. After a trade-off between performance improvement and computational cost growth, we choose 4, 8, 2 as the value of H, S, and N.

B. The Discussion of Different Motion Field

To verify the effectiveness of consistent motion generated by the consistent motion learning component that described in Sec. III-B1. As shown in Fig. 8, we compare the visual qualities of consistent motion with other state-of-the-art flow-based algorithms, such as BMBC [36] and ABME [37]. The consistent motion has clearer textures, which indicates the superiority of the generated consistent motion field.

C. The Discussion of Multi-Scale Fusion Structure

In the previous works [26], [52], stacking transformers in multi-layer and multi-scale has been proven to be effective. Therefore, as described in Sec. III-D3 and shown in Fig. 4, we stack the trajectory-aware Transformer at multi-scale (i.e. $\times1$, $\times2$, and $\times4$), and each scale contains multiple layers of attention mechanisms. In this section, we investigate the effects of Transformer multi-scale stacking on performance. As shown in Tab. VI, the features of multi-scale can effectively facilitate the interaction of multi-scale features and improve the performance. This demonstrates the effectiveness of feature fusion and interaction in multi-scale.

D. The Discussions of Perceptual Quality

We use LPIPS [55] as a widely used metric to evaluate perceptual quality. Results, shown in the following Additional Tab. VII, demonstrate that TTVFI is comparable with the state-of-the-art VFIformer [28] and still highly superior in the perceptual metrics.

E. The Discussions of Multi-Frame Input

Compared to the two-frame input algorithm framework in this paper, many existing works [20], [42], [50] utilize more frames (i.e., four frames) as network inputs to obtain higher

TABLE VII
PERCEPTUAL QUALITY COMPARISON (LPIPS↓) ON THE VIMEO-90K [51]

Method	MEMC [2]	DAIN [1]	AdaCoF [23]	BMBC [36]	EDSC [6]	ABME [37]	RIFE-Large [16]	VFIformer [28]	**TTVFI**
LPIPS	0.0256	0.0224	0.0304	0.0235	0.0263	0.0212	0.0227	0.0202	0.0205

TABLE VIII
QUANTITATIVE COMPARISON (PSNR↑ AND SSIM↑) ON THE VIMEO-90K [51] AND DAVIS [39] DATASETS USING MULTIPLE FRAMES

Method	#Params(M)	Vimeo-90K		DAVIS	
		PSNR	SSIM	PSNR	SSIM
QVI [50]	29.2	35.15	0.971	27.17	0.874
SoftSplat [32]	7.7	35.76	0.972	27.42	0.878
FLAVR [20]	42.4	36.30	0.975	27.44	0.874
VFIF-S [42]	7.5	36.48	0.976	27.92	0.885
VFIF-S [42]	29.0	36.96	0.978	28.09	0.888
TTVFI$_m$	25.6	36.87	0.978	28.19	0.889

TABLE IX
QUANTITATIVE COMPARISON (PSNR↑ AND SSIM↑) ON THE ULTRA-LARGE VIDEO INTERPOLATION DATASET X4K1000FPS [43]

Method	#Params(M)	X4K1000FPS	
		PSNR(dB)↑	SSIM↑
DAIN [1]	24.0	26.78	0.8065
AdaCoF [23]	22.9	23.90	0.7271
XVFI [43]	5.5	30.12	0.8704
ABME [37]	18.1	30.16	0.8793
TTVFI	16.6	30.51	0.8850

TABLE X
QUANTITATIVE COMPARISON (PSNR↑ AND SSIM↑) OF BLURRY VIDEO INTERPOLATION FRAMES ON THE ADOBE240 [45] DATASET

Method	#Params(M)	Adobe240	
		PSNR(dB)↑	SSIM↑
TNTT [19]	10.8	29.24	0.8754
BIN [40]	4.68	32.51	0.9280
PRF [41]	11.4	33.31	0.9372
DeMFI [35]	5.96	33.93	0.9441
TTVFI$_m$	25.6	33.42	0.9383

quality interpolation results. To verify the effectiveness of our trajectory transformer when utilizing more frames from farther distances, we extend our TTVFI to multi-frame input as well and added the comparisons of the latest multi-frame input-based video interpolation methods. Specifically, we predict the optical flow between multiple frames by the motion estimation network in this part, where the trajectories are computed by Eq. (4) in the main paper. By doing so, the number of warped features and extracted features in the Sec. III-C are doubled. Correspondingly, we increase the number of input channels in the attention mechanism to adapt to the doubled number of features. Notably, for fair comparisons, the same data set settings as the existing methods [20], [42] have been used. As shown in Tab. VIII, compared to VFIT [42], our TTVFI achieves comparable performance using fewer parameters and exceeds 0.1 dB on the DAVIS dataset with complex motion. Such superior performances mainly benefit from the use of trajectories in attention calculation.

F. The Discussions of High-Resolution Frame Input

Higher resolution video tends to contain larger and more complex motion. To validate the interpolation performance of our method at higher input resolutions and more complex motions, we follow existing works [37], [43] and construct comparisons with other methods on the ultra-large 4K high-resolution dataset X4K1000FPS [43]. As shown in Tab. IX, due to the recursive structure used in XVFI [43] to share the parameters in each scale, our method has a larger number of parameters. Nevertheless, our TTVFI still significantly outperforms ABME [37] by 0.35 dB at high-resolution input. This demonstrates that the proposed trajectory can effectively alleviate the effects of inconsistent warping at complex motion with high-resolution input.

G. The Discussions of Blurry Frame Input

Low frame-rate videos are often associated with degradation in visual quality, and we further use blurry video frame interpolation benchmark to verify the generalization ability of the proposed TTVFI. We follow existing works [40], [41], [45] to use 8 videos of the Adobe240 dataset [45] and construct comparisons with other methods. It is worth noting that, compared to the blurry video interpolation task, our method focuses more on the accuracy of motion information rather than on blur removal. As shown in Tab. X, due to the frame blur, our method produces inaccurate trajectories, which leads to performance degradation. Besides, to generate more accurate trajectories, our method includes a motion estimation network and a consistency motion learning component, which inevitably introduces additional parameters. Nevertheless, our TTVFI still achieves a higher performance than BIN [40] and RFP [41]. This proves that our TTVFI is robust in blurry video frame interpolation.

VI. LIMITATIONS

In this section, we discuss the limitations of TTVFI and the failure cases as shown in Fig. 9.

A. Rotation

Although we propose a consistent motion learning component to generate consistent motion, when facing complex motion (e.g., rotation), as shown in the upper part of Fig. 9, the accuracy of the motion trajectory is limited and the significance is reduced.

B. Camera Motion

We propose to pay more attention to the regions with inconsistent motion, which usually focuses on the moving instances in the frame. However, when intense camera motion

Fig. 9. Failure case when rotation and camera motion occur.

occurs, the motion of frame boundaries is inconsistent and incomplete. As shown in the bottom part of Fig. 9, the incomplete motion makes little information that can be used to recover frames boundaries, leading to poor results.

VII. Conclusion

In this paper, we pay more attention to the important synthesis network in VFI and propose a new trajectory-aware transformer (TTVFI). In particular, TTVFI aims to mitigate the distortion and blur caused by inconsistent motion and inaccurate warping in existing algorithms, and learns more accurate features of the intermediate frames from the original input frames. To implement such formulations better, we first propose a consistent motion learning component to generate the consistent motion field, which can be defined as a group of inter-frame motion trajectories. Then we formulate video frames into two kinds of pre-aligned visual tokens and calculate attention separately according to whether the regional motion is consistent or not. To our best knowledge, TTVFI is the first work to enables Transformers to model the features of intermediate frames by motion trajectory in VFI. Experimental results show the superiority between the proposed TTVFI and existing SOTA methods. In the future, we will focus on extending transferring the trajectory-aware Transformer in more low-level vision tasks by more explorations.

References

[1] W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, and M.-H. Yang, "Depth-aware video frame interpolation," in *Proc. CVPR*, Jun. 2019, pp. 3703–3712.

[2] W. Bao, W.-S. Lai, X. Zhang, Z. Gao, and M.-H. Yang, "MEMC-Net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 933–948, Mar. 2021.

[3] W. Bao, X. Zhang, L. Chen, L. Ding, and Z. Gao, "High-order model and dynamic filtering for frame rate up-conversion," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3813–3826, Aug. 2018.

[4] J. Cao, Y. Li, K. Zhang, and L. Van Gool, "Video super-resolution transformer," 2021, *arXiv:2106.06847*.

[5] X. Cheng and Z. Chen, "Video frame interpolation via deformable separable convolution," in *Proc. AAAI*, vol. 34, 2020, pp. 10607–10614.

[6] X. Cheng and Z. Chen, "Multiple video frame interpolation via enhanced deformable separable convolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 7029–7045, Oct. 2022.

[7] M. Choi, H. Kim, B. Han, N. Xu, and K. M. Lee, "Channel attention is all you need for video frame interpolation," in *Proc. AAAI*, vol. 34, 2020, pp. 10663–10671.

[8] J. Dai et al., "Deformable convolutional networks," in *Proc. ICCV*, Oct. 2017, pp. 764–773.

[9] D. Danier, F. Zhang, and D. Bull, "FloLPIPS: A bespoke video quality metric for frame interpoation," 2022, *arXiv:2207.08119*.

[10] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[11] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, "Deep stereo: Learning to predict new views from the world's imagery," in *Proc. CVPR*, Jun. 2016, pp. 5515–5524.

[12] D. Fourure, R. Emonet, E. Fromont, D. Muselet, A. Tremeau, and C. Wolf, "Residual conv-deconv grid network for semantic segmentation," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 181.1–181.13.

[13] S. Gui, C. Wang, Q. Chen, and D. Tao, "FeatureFlow: Robust video interpolation via structure-to-texture generation," in *Proc. CVPR*, Jun. 2020, pp. 14004–14013.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. ICCV*, Dec. 2015, pp. 1026–1034.

[15] Q. Hou, A. Ghildyal, and F. Liu, "A perceptual quality metric for video frame interpolation," in *Proc. ECCV*. Cham, Switzerland: Springer, 2022, pp. 234–253.

[16] Z. Huang, T. Zhang, W. Heng, B. Shi, and S. Zhou, "Real-time intermediate flow estimation for video frame interpolation," in *Proc. ECCV*. Cham, Switzerland: Springer, 2022, pp. 624–642.

[17] X. Jia, B. D. Brabandere, T. Tuytelaars, and L. Van Gool, "Dynamic filter networks," in *Proc. NeurIPS*, vol. 29, 2016, pp. 1–9.

[18] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. Kautz, "Super SloMo: High quality estimation of multiple intermediate frames for video interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9000–9008.

[19] M. Jin, Z. Hu, and P. Favaro, "Learning to extract flawless slow motion from blurry videos," in *Proc. CVPR*, Jun. 2019, pp. 8112–8121.

[20] T. Kalluri, D. Pathak, M. Chandraker, and D. Tran, "FLAVR: Flow-agnostic video representations for fast frame interpolation," in *Proc. WACV*, Jan. 2023, pp. 2071–2082.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–15.

[22] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep Laplacian pyramid networks for fast and accurate super-resolution," in *Proc. CVPR*, Jul. 2017, pp. 624–632.

[23] H. Lee, T. Kim, T.-Y. Chung, D. Pak, Y. Ban, and S. Lee, "AdaCoF: Adaptive collaboration of flows for video frame interpolation," in *Proc. CVPR*, Jun. 2020, pp. 5316–5325.

[24] C. Liu, H. Yang, J. Fu, and X. Qian, "Learning trajectory-aware transformer for video super-resolution," in *Proc. CVPR*, Jun. 2022, pp. 5687–5696.

[25] Y.-L. Liu, Y.-T. Liao, Y.-Y. Lin, and Y.-Y. Chuang, "Deep video frame interpolation using cyclic frame generation," in *Proc. AAAI*, vol. 33, 2019, pp. 8794–8802.

[26] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.

[27] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," in *Proc. ICCV*, Oct. 2017, pp. 4463–4471.

[28] L. Lu, R. Wu, H. Lin, J. Lu, and J. Jia, "Video frame interpolation with transformer," in *Proc. CVPR*, Jun. 2022, pp. 3532–3542.

[29] S. Meister, J. Hur, and S. Roth, "UnFlow: Unsupervised learning of optical flow with a bidirectional census loss," in *Proc. AAAI*, vol. 32, no. 1, 2018.

[30] X. Min, G. Zhai, J. Zhou, M. C. Q. Farias, and A. C. Bovik, "Study of subjective and objective quality assessment of audio-visual signals," *IEEE Trans. Image Process.*, vol. 29, pp. 6054–6068, 2020.

[31] S. Niklaus and F. Liu, "Context-aware synthesis for video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1701–1710.

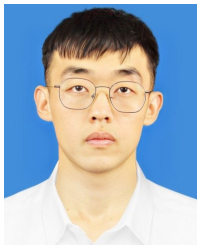
[32] S. Niklaus and F. Liu, "Softmax splatting for video frame interpolation," in *Proc. CVPR*, Jun. 2020, pp. 5437–5446.

[33] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive convolution," in *Proc. CVPR*, Jul. 2017, pp. 670–679.

[34] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive separable convolution," in *Proc. ICCV*. Cham, Switzerland: Springer, Oct. 2017, pp. 261–270.

[35] J. Oh and M. Kim, "DeMFI: Deep joint deblurring and multi-frame interpolation with flow-guided attentive correlation and recursive boosting," in *Proc. ECCV*. Glasgow, U.K.: Springer, Aug. 2022, pp. 198–215.

[36] J. Park, K. Ko, C. Lee, and C.-S. Kim, "BMBC: Bilateral motion estimation with bilateral cost volume for video interpolation," in *Proc. ECCV*. Springer, 2020, pp. 109–125.
[37] J. Park, C. Lee, and C.-S. Kim, "Asymmetric bilateral motion estimation for video frame interpolation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 14539–14548.
[38] M. Patrick et al., "Keeping your eye on the ball: Trajectory attention in video transformers," *NeurIPS*, vol. 34, 2021, pp. 12493–12506.
[39] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proc. CVPR*, Jun. 2016, pp. 724–732.
[40] W. Shen, W. Bao, G. Zhai, L. Chen, X. Min, and Z. Gao, "Blurry video frame interpolation," in *Proc. CVPR*, Jun. 2020, pp. 5114–5123.
[41] W. Shen, W. Bao, G. Zhai, L. Chen, X. Min, and Z. Gao, "Video frame interpolation and enhancement via pyramid recurrent framework," *IEEE Trans. Image Process.*, vol. 30, pp. 277–292, 2021.
[42] Z. Shi, X. Xu, X. Liu, J. Chen, and M.-H. Yang, "Video frame interpolation transformer," in *Proc. CVPR*, Jun. 2022, pp. 17482–17491.
[43] H. Sim, J. Oh, and M. Kim, "XVFI: Extreme video frame interpolation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 14489–14498.
[44] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*.
[45] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang, "Deep video deblurring for hand-held cameras," in *Proc. CVPR*, Jul. 2017, pp. 1279–1288.
[46] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. CVPR*, Jun. 2018, pp. 8934–8943.
[47] A. Vaswani et al., "Attention is all you need," in *Proc. NeurIPS*, 2017, pp. 5998–6008.
[48] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[49] C.-Y. Wu, N. Singhal, and P. Krahenbuhl, "Video compression through image interpolation," in *Proc. ECCV*, 2018, pp. 416–431.
[50] X. Xu, L. Siyao, W. Sun, Q. Yin, and M.-H. Yang, "Quadratic video interpolation," in *Proc. NeurIPS*, vol. 32, 2019, pp. 1–10.
[51] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *Int. J. Comput. Vis.*, vol. 127, no. 8, pp. 1106–1125, Aug. 2019.
[52] F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo, "Learning texture transformer network for image super-resolution," in *Proc. CVPR*, Jun. 2020, pp. 5791–5800.
[53] Y. Zeng, J. Fu, and H. Chao, "Learning joint spatial–temporal transformations for video inpainting," in *Proc. ECCV*, 2020, pp. 528–543.
[54] G. Zhai and X. Min, "Perceptual image quality assessment: A survey," *Sci. China Inf. Sci.*, vol. 63, no. 11, pp. 1–52, Nov. 2020.
[55] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. CVPR*, Jun. 2018, pp. 586–595.
[56] L. Zhao, S. Wang, X. Zhang, S. Wang, S. Ma, and W. Gao, "Enhanced motion-compensated video coding with deep virtual reference frame generation," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 4832–4844, Oct. 2019.
[57] Z. Zhong, M. Cao, X. Ji, Y. Zheng, and I. Sato, "Blur interpolation transformer for real-world motion from blur," 2022, *arXiv:2211.11423*.
[58] Z. Zhong, X. Sun, Z. Wu, Y. Zheng, S. Lin, and I. Sato, "Animation from blur: Multi-modal blur decomposition with motion guidance," in *Proc. ECCV*. Cham, Switzerland: Springer, 2022, pp. 599–615.
[59] Y. Zou, Z. Luo, and J.-B. Huang, "DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency," in *Proc. ECCV*, 2018, pp. 36–53.



Chengxu Liu received the B.E. degree from Xi'an Jiaotong University, Xi'an, China, in 2019. He was an Intern with the Multimedia Search and Mining Group, Microsoft Research Asia, from April 2021 to May 2022. He is currently pursuing the Ph.D. degree with the SMILES Laboratory, Xi'an Jiaotong University. His current research interests include fine-grained image classification, object detection, video super-resolution, video frame interpolation, and image enhancement.

Huan Yang received the B.S. and Ph.D. degrees in computer science from Shanghai Jiao Tong University, China, in 2014 and 2019, respectively. He is currently a Senior Researcher with Microsoft Research Asia, Beijing, China. His current research interests include computer vision, image processing, real-time video processing, and image photography.

Jianlong Fu received the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Science, in 2015. He is currently a Senior Research Manager with the Multimedia Search and Ming Group, Microsoft Research Asia, Beijing, China. He has shipped core technologies to Microsoft products, including windows photo, Bing image search, Xiaoice, Chatbot, and Microsoft flower. His current research interests include computer vision and multimedia content analysis, especially on fine-grained image recognition, vision and language, and personal photo experience of browsing, searching, sharing, and storytelling.

Xueming Qian (Member, IEEE) received the B.S. and M.S. degrees from the Xi'an University of Technology, Xi'an, China, in 1999 and 2004, respectively, and the Ph.D. degree in electronics and information engineering from Xi'an Jiaotong University, Xi'an, in 2008.

From 2011 to 2014, he was an Associate Professor with Xi'an Jiaotong University, where he is currently a Full Professor and the Director of the SMILES Laboratory. He was a Visiting Scholar with Microsoft Research Asia, Beijing, China, from 2010 to 2011. His current research interests include social media big data mining and search.

Prof. Qian was a recipient of the Microsoft Fellowship in 2006 and the Outstanding Doctoral Dissertations of Xi'an Jiaotong University and Shaanxi Province in 2010 and 2011, respectively.