electronics

MDPI

Article

Parallel Spatio-Temporal Attention Transformer for Video Frame Interpolation

Xin Ning [1,*], Feifan Cai [1], Yuhang Li [1] and Youdong Ding [1,2]

[1] College of Shanghai Film, Shanghai University, 788 Guangzhong Road, Shanghai 200072, China; feifan@shu.edu.cn (F.C.); yuhangli@shu.edu.cn (Y.L.); ydding@shu.edu.cn (Y.D.)
[2] Shanghai Engineering Research Center of Motion Picture Special Effects, 788 Guangzhong Road, Shanghai 200072, China
* Correspondence: innerpeacenx@shu.edu.cn

Abstract: Traditional video frame interpolation methods based on deep convolutional neural networks face challenges in handling large motions. Their performance is limited by the fact that convolutional operations cannot directly integrate the rich temporal and spatial information of inter-frame pixels, and these methods rely heavily on additional inputs such as optical flow to model motion. To address this issue, we develop a novel framework for video frame interpolation that uses Transformer to efficiently model the long-range similarity of inter-frame pixels. Furthermore, to effectively aggregate spatio-temporal features, we design a novel attention mechanism divided into temporal attention and spatial attention. Specifically, spatial attention is used to aggregate intra-frame information, integrating both attention and convolution paradigms through the simple mapping approach. Temporal attention is used to model the similarity of pixels on the timeline. This design achieves parallel processing of these two types of information without extra computational cost, aggregating information in the space–time dimension. In addition, we introduce a context extraction network and multi-scale prediction frame synthesis network to further optimize the performance of the Transformer. Our method and state-of-the-art methods are extensively quantitatively and qualitatively experimented on various benchmark datasets. On the Vimeo90K and UCF101 datasets, our model achieves improvements of 0.09 dB and 0.01 dB in the PSNR metrics over UPR-Net-large, respectively. On the Vimeo90K dataset, our model outperforms FLAVR by 0.07 dB, with only 40.56% of its parameters. The qualitative results show that for complex and large-motion scenes, our method generates sharper and more realistic edges and details.

Keywords: video frame interpolation; spatio-temporal attention mechanism; Transformer; multi-scale information

check for updates

Citation: Ning, X.; Cai, F.; Li, Y.; Ding, Y. Parallel Spatio-Temporal Attention Transformer for Video Frame Interpolation. *Electronics* **2024**, *13*, 1981. https://doi.org/10.3390/electronics13101981

Academic Editor: Dah-Jye Lee

Received: 2 April 2024
Revised: 7 May 2024
Accepted: 14 May 2024
Published: 18 May 2024

1. Introduction

Video frame interpolation (VFI) aims to improve the frame rate of a video by synthesizing new intermediate frames between consecutive frames on the timeline. This task has been widely used in the fields of video compression [1], video enhancement [2,3], and slow motion generation [4].

Currently, most popular VFI methods are mainly based on convolutional neural networks (CNNs) [5–8]. Although these methods achieve remarkable performance, they exhibit obvious limitations in handling large motions in complex scenes. Specifically, the CNN-based methods generally rely on extra optical flow [9] warping to model inter-frame motion [5,6]. Despite this approach being effective in handling linear motion, it faces many challenges for complex nonlinear motion estimation. Therefore, this limits the ability of CNN-based methods to handle large motion and increases the significant computational cost [5], which prevents the further development and optimization of VFI models. In addition, CNNs are less capable of capturing inter-frame long-range information due to their restricted receptive fields, and using larger kernels increases computational overhead and model parameters.

Recently, Transformer performed well in several tasks in computer vision [10–13]. It has a flexible architecture that can effectively capture the long-range dependencies of pixels and overcome the drawbacks of CNNs mentioned above. Thus, the architecture is well suited for VFI tasks. However, it remains a challenging problem to apply Transformer to VFI tasks and capture information on the timeline while aggregating video spatial information.

In light of these challenges, this paper proposes a new VFI model based on Transformer architecture for synthesizing realistic video frames. Specifically, Transformer relies on the self-attention mechanism to capture long-range information between pixels. The computational complexity of this mechanism is proportional to the number of input pixels, and thus directly applying it to video data leads to extremely high computational cost. In addition, some Transformer-based methods [14] only interact with pixels of a single image globally, and are unable to directly adapt to the time dimension of video frames. To address the two issues, we design the parallel spatio-temporal attention (PSTA) mechanism using a parallel strategy, which is divided into temporal attention (TA) and spatial attention (SA), dedicated to modeling pixel similarity in the time dimension and aggregating spatial information of intra-frame pixels, respectively. In SA, in order to enhance the fine-grained dependencies between intra-frame pixels, we design the SA as a mixture of two paradigms, convolution and self-attention. Furthermore, to reduce the computational complexity, we employ the simple mapping approach to process the input features so that they can be used as inputs for both paradigms at the same time.

Second, while Transformer achieves the information interaction between remote pixels, in order to avoid losing the pixel information of the original frames and preserve more texture details, we propose two sub-networks: context extraction network (CE-Net) and multi-scale prediction frame synthesis network (MPFS-Net). CE-Net is devoted to preserving the rich detailed information of the input frames, and MPFS-Net is able to fuse the structure and information of video frames at different scales to synthesize high-quality intermediate frames.

Our contributions are summarized as follows:

1. We propose a novel Transformer-based VFI framework. It overcomes the limitations of traditional CNN-based methods and can effectively model the long-range dependencies between pixels.
2. We design a new attention mechanism, PSTA. It is divided into TA and SA, and the mechanism can process inter-frame spatio-temporal information in parallel to efficiently process video frames. TA captures inter-frame pixel temporal variations, and SA efficiently aggregates spatial features. SA is designed as the combination of both convolutional and self-attention paradigms, and the input features are processed by a simple mapping approach to suit both paradigms, which improves the quality and realism of the synthesized frames. We also propose two sub-networks, CE-Net and MPFS-Net, for enhancing the details of synthesized frames and fusing the information of multi-scale video frames, respectively.
3. Our model demonstrates significant performance on various benchmark datasets, with higher processing efficiency and fewer parameters. As shown in Figure 1, our standard model (Ours) outperforms the state-of-the-art (SOTA) methods ABME [15] and FLAVR [16] by 0.19 dB and 0.07 dB, respectively, with only 95.02% and 40.56% of their parameters, respectively.

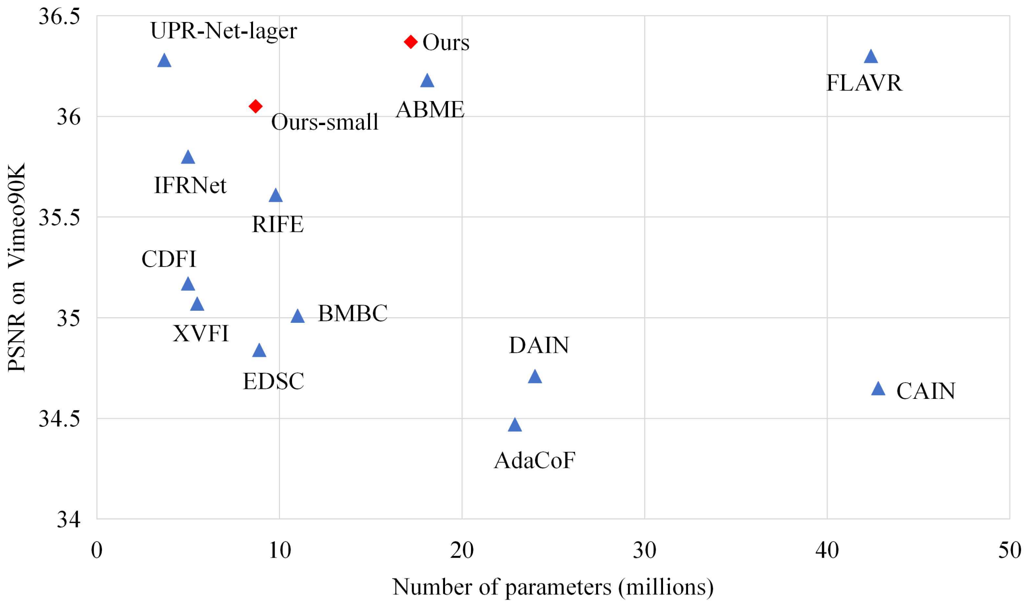


Figure 1. Comparison of model size and performance with state-of-the-art methods using the Vimeo90K [8] dataset. Our method achieves an ideal balance between excellent performance and model parameters.

2. Related Work

2.1. Video Frame Interpolation

The existing VFI methods can be mainly classified into two categories: flow-based [5,8,17,18] and kernel-based [19–22].

In flow-based VFI methods [5,6,18], input frames are warped by estimating the optical flow to synthesize the intermediate frames. Niklaus and Liu [18] design a SoftSplat model based on optical flow and a feature pyramid, which uses softmax splatting for forward warping. Bao et al. [5] propose DAIN, which uses depth information to explicitly detect occlusions and employs optical flow with a local interpolation kernel to warp the intermediate flow. However, the complexity of optical flow estimation significantly affects the processing speed of the model. To address this issue, Huang et al. [6] propose RIFE, a VFI model based on privileged distillation, which is able to improve the processing speed by estimating the intermediate flow in real time. Although the above methods are effective in scenes dealing with simple motions, inaccuracies in optical flow estimation can limit the performance of the model when dealing with complex nonlinear motions.

Kernel-based VFI methods [7,19–25] do not rely on any predefined assumptions, and they generate new frames by using CNN estimation of spatially adaptive convolution kernels. Therefore, kernel-based methods can overcome the drawback of inaccurate optical flow estimation and have been widely used in various videos. Niklaus et al. [23] propose adaptive separable convolution, replacing the original 2D convolution kernel with a pair of 1D convolution kernels, which reduces the number of operations and the number of parameters of the model. In video processing tasks, deformable convolution (DConv) has been shown to enhance the flexibility of network encoders [26]. Inspired by DConv [27], Lee et al. [7] propose AdaCoF, a model with a learned deformable spatial convolution kernel, which solves the problem of limited degrees of freedom for ordinary convolution kernels. Cheng and Chen [24] propose DSepConv, which uses deformable separated convolution to extend the kernel-based approach, and further propose EDSC [25] for multi-frame interpolation. To reduce model parameters, Ding et al. [19] propose CDFI, a compression-driven VFI-based model. The model compresses AdaCoF by model pruning and adds multi-scale details. Zhang et al. [20] propose a local lightweight strategy based on a bidirectional encoding structure with a channel attention cascade and a VFI network, L^2BEC^2. This strategy not only improves the visual quality but can also be migrated into the

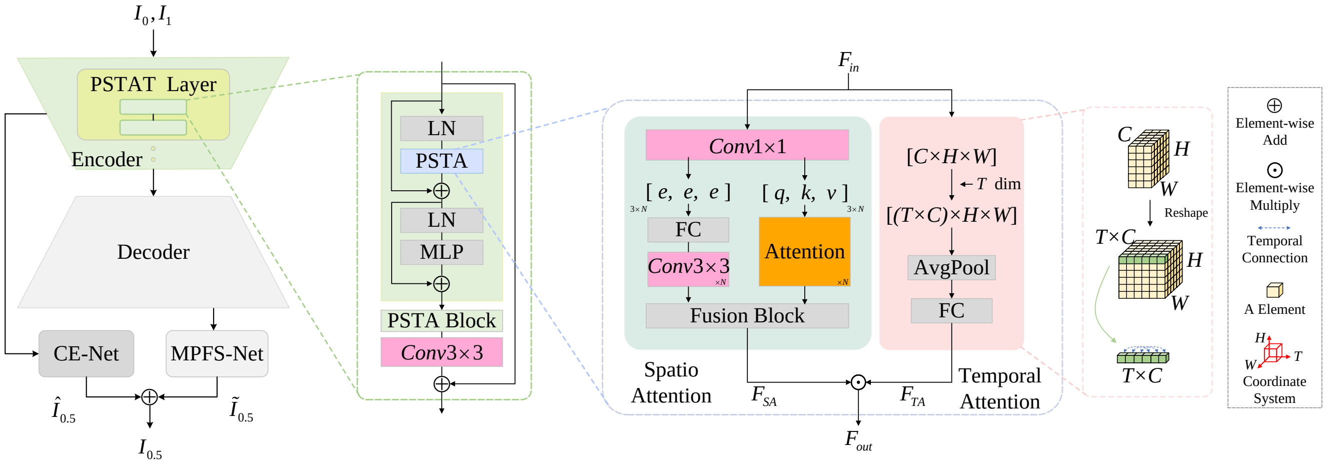
AdaCoF model, thus effectively reducing its number of parameters. Ding et al. [21] propose a unified warping framework named MSEConv. The authors introduce an occlusion masking operation to enhance the robustness of motion occlusion. Overall, kernel-based methods typically employ a fixed-size convolutional kernel for prediction, which limits their effectiveness in handling fine-grained features of video frames and motion information at different scales. In addition, these methods also fail to adequately consider the long-range dependence of inter-frame pixels in the time dimension. In contrast to these earlier SOTA methods, we propose a novel Transformer-based VFI model that does not rely on external inputs and can effectively simulate large motion in real scenes.

2.2. Vision Transformer

Due to its flexibility and high performance, Transformer [28] is widely used in computer vision [10]. Carion et al. [13] propose a model for end-to-end target detection, named DETR. Liang et al. [29] propose SwinIR, an image recovery model based on the Swin Transformer [10]. While Transformer performs well in some image tasks, it is not directly applicable to video. Recently, some researchers [14,30] have explored applying Transformer to VFI tasks. For example, Lu et al. [14] propose a network based on a cross-scale window attention mechanism, VFIformer. However, the approach fails to extend the attention mechanism to the time dimension of the input frames, and only works on a single image. In contrast, we propose a parallel scheme that is able to apply both kinds of attention in the spatio-temporal domain without sacrificing processing efficiency, thus effectively aggregating spatio-temporal information.

3. Proposed Method

The architecture of our proposed model is shown in Figure 2a and contains three main parts: the Transformer-based encoder–decoder architecture, the CE-Net, and the MPFS-Net. In particular, the encoder consists of four parallel spatio-temporal attention Tansformer (PSTAT) layers, and each PSTAT layer contains two Transformer residual blocks (TRBs). As shown in Figure 2b, each TRB consists of two PSTA blocks and a convolution. The PSTA block consists of the PSTA mechanism, layer normalization (LN), and multi-layer perceptron (MLP). The LN and the residual link help to stabilize the training, and the MLP uses a two-layer structure with activation using the GELU function [31].



(a) Model Architecture (b) Transformer Residual Block (c) PSTA (d) Temporal Attention

Figure 2. The overall architecture of our proposed method. (**a**) Architecture of our model; (**b**) Transformer residual block structure in the parallel spatio-temporal attention Transformer layer; (**c**) Parallel spatio-temporal attention; (**d**) Temporal attention dimension transformation.

The aim of this study is to generate the middle frame $I_{0.5}$ between consecutive input frames I_0 and I_1. Firstly, I_0 and I_1 are passed through the encoder to obtain the corresponding output F_i^{Enc} $(i = 0, 1, 2, 3)$ for each PSTAT layer, to extract the intra-frame and inter-frame features. The decoder contains three 2D deconvolution upsampling layers with a stride of 2, and the output of each layer is denoted as F_i^{Dec} $(i = 0, 1, 2)$. F_i^{Enc} and F_i^{Dec} flow to CE-Net and MPFS-Net, respectively, to generate the multi-scale intermediate frames $\hat{I}_{0.5}$ and $\tilde{I}_{0.5}$, respectively. Finally, they are element-wise added to obtain the middle frame $I_{0.5}$. The architecture of the model can be formally represented as:

$$\begin{aligned}
F_i^{Enc} &= Enc(PSTAT^i([I_0, I_1])),\\
F_i^{Dec} &= Dec(ConvTranspose2d^i(F_i^{Enc})),\\
\hat{I}_{0.5} &= CENet(F_i^{Enc}),\\
\tilde{I}_{0.5} &= MPFSNet(F_i^{Dec}),\\
I_{0.5} &= \hat{I}_{0.5} + \tilde{I}_{0.5}
\end{aligned} \tag{1}$$

3.1. Parallel Spatio-Temporal Attention

In computer vision, although Transformer shows exceptional performance in image processing tasks [10,32] through the attention mechanism and shift windows, this approach is not fully applicable to video data. For the VFI task, it is crucial to process large inter-frame motions and aggregate the temporal information between two frames. Therefore, we propose a parallel attention mechanism-PSTA that simultaneously processes intra-frame spatial information and inter-frame temporal information, as shown in Figure 2c. Specifically, the input feature tensor and output feature tensor are denoted as $F_{in} \in \mathbb{R}^{C\times H\times W}$ and $F_{out} \in \mathbb{R}^{C\times H\times W}$, with C, H, W being channel, height, and width, respectively. We first pass the $F_{in} \in \mathbb{R}^{C\times H\times W}$ to SA and TA, respectively. SA is used to aggregate the intra-frame spatial features of the input frames, and TA is used to capture the temporal variations of the inter-frame pixels, modeling their similarity in the time dimension. After extracting the valid features in parallel, these two types of features are combined by element-wise multiplication to form the output feature $F_{out} \in \mathbb{R}^{C\times H\times W}$. This allows the model to simultaneously learn the spatio-temporal information of the pixels. The computational process is as follows:

$$\begin{aligned}
F_{SA} &= SA(F_{in}),\\
F_{TA} &= TA(F_{in}),\\
F_{out} &= F_{SA} \odot F_{TA}
\end{aligned} \tag{2}$$

where $\odot$ is the Hadamard product, the detailed computation of F_{SA} and F_{TA} is described in Sections 3.1.1 and 3.1.2. Next, we will describe the structure of PSTA in detail.

3.1.1. Spatio Attention

In order to efficiently extract intra-frame spatial features, SA is designed as a combination of self-attention mechanism and convolution as shown in Figure 2c. The SA module mainly contains three stages: feature mapping, feature extraction, and feature fusion. Firstly, in the feature mapping stage, we use three 1 × 1 convolutions to map and reshape the input feature $F_{in} \in \mathbb{R}^{C\times H\times W}$ into 3 × N intermediate feature blocks. These intermediate feature blocks can be shared for both self-attention and convolution operations and are represented as N sets of query, key, and value (all with $\in \mathbb{R}^{C/N\times H\times W}$) for self-attention and N sets of convolution elements $e \in \mathbb{R}^{C/N\times H\times W}$ for convolution operations. Intermediate feature sharing avoids additional computation and simplifies the overall structure of the SA.

During the feature extraction stage, intermediate feature blocks are processed according to different paradigms. For the convolution operation, 3 × N intermediate feature blocks are reshaped into N feature maps $f_{in}^{conv} \in \mathbb{R}^{C/N\times H\times W}$ by a fully connected (FC) layer, and then features are extracted from each set of feature maps by the convolution operation with a kernel size of 3 × 3, and the resulting N sets of output features are denoted as $f_{out}^{conv} \in \mathbb{R}^{C/N \times H \times W}$. For the self-attention operation, we use the standard multi-head self-attention mechanism for feature aggregation. Specifically, for pixel $p(i,j)$, its corresponding input feature tensor is $f_{ij}^{att} \in \mathbb{R}^{C/N}$. Each set of intermediate feature blocks after mapping is directly used as query, key, and value. W_q, W_k, W_v are the projection matrices of query, key, and value, and their matrices can be expressed as:

$$q_{ij} = W_q f_{ij}^{att}, k_{ij} = W_k f_{ij}^{att}, v_{ij} = W_v f_{ij}^{att} \tag{3}$$

Next, we aggregate the local features of the pixel by performing a self-attention operation on the ×3 intermediate feature blocks (query, key, value). The standard self-attention computation involves two main steps: calculating attention weights and aggregating value matrices. In this paper, we combine these two steps into one, and the specific computation process of the attention is as follows:

$$Attention(q_{ij}, k_{mn}, \nu_{mn}) = \underset{m,n \in P}{\text{softmax}}(\frac{q_{ij}^T k_{mn}}{\sqrt{d}}) v_{mn} \tag{4}$$

where d is the feature dimension of q_{ij}. The corresponding acceptance domain of the query is denoted as $P(i,j)$. In the third stage, we concatenate the N sets of outputs produced by each of the two operations. Subsequently, they are fused by addition, where the intensity of the convolutional output is controlled by the parameter λ. Finally, the output $F_{SA} \in \mathbb{R}^{C \times H \times W}$ of the SA module can be expressed by the following equation:

$$F_{SA} = F_{att} + \lambda F_{conv} \tag{5}$$

3.1.2. Temporal Attention

Despite spatial attention effectively extracting the information within individual frames, it fails to focus on the temporal information and variations between input frames. Moreover, in order to reduce the model parameters and computational cost, inspired by [33], we design a simple temporal attention mechanism to enhance the sensitivity and adaptability of the model to temporal changes, as shown in Figure 2c. Specifically, for the input tensor $F_{in} \in \mathbb{R}^{C \times H \times W}$, we add a time dimension T, reshaping $C \times H \times W$ as $T \times C \times H \times W$. Then, we combine the channel and time dimensions, reshaping it to $(T \times C) \times H \times W$. The purpose of this is to sequentially arrange the channels of the two frames in the temporal domain, facilitating the model to learn the information on the timeline.

Next, an average pooling (AvgPool) is used to perform the squeeze operation on the reshaped tensor F'_{in}, which is an aggregation strategy that encodes features in the $(T \times C)$ dimension as a global feature. This is followed by the FC layer that fuses information from different channel feature maps. Then, a sigmoid function is used to map and obtain attention weights of dimension $(T \times C)$. Finally, we remove the time dimension T from F_{TA} and reshape the feature tensor back to $C \times 1 \times 1$, multiplying the output with SA to obtain $F_{out} \in \mathbb{R}^{C \times H \times W}$. As shown in Figure 2d, this approach focuses on the temporal variation of pixels between frames, explicitly modeling the correlation of pixels in the time dimension and enabling the model to learn the temporal information between frames. The process of TA calculation is summarized below:

$$\begin{aligned} F'_{in} &= \text{Reshape}(F_{in}, [T, C, H, W]), \\ F'_{TA} &= \text{FC}(\text{AvgPool}(F'_{in})), \\ F_{TA} &= \text{Reshape}(F'_{TA}, [C, H, W]) \end{aligned} \tag{6}$$

where $F'_{TA} \in \mathbb{R}^{(T \times C) \times 1 \times 1}$, $F_{TA} \in \mathbb{R}^{C \times 1 \times 1}$, and Reshape are tensor dimension reshaping operations.

3.1.3. Computational Cost

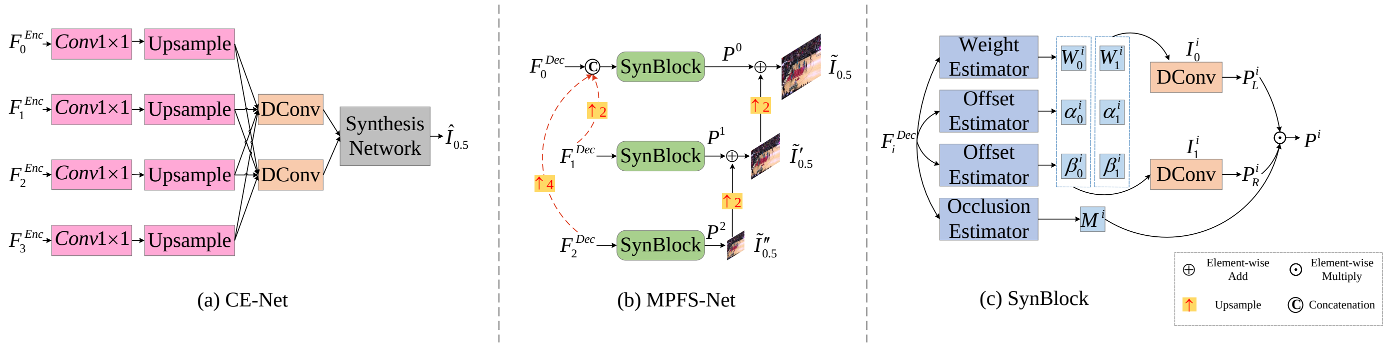
To clearly demonstrate the computational overhead of our model, we analyze the floating point operations (FLOPs) of the SA and TA modules in detail. The results are presented in Table 1, for SA, the computational overhead of the simple mapping stage is $O(3C^2HW)$, which has quadratic complexity with the number of channels C. In the feature aggregation stage, both the convolutional operation and the attention mechanism are linear to C, and their computational overheads are $O(K^2C)$ and $O(3CK^2)$, respectively. This indicates that the main computational overhead of SA is concentrated in the simple mapping stage. The computational overhead for TA overall similarly has quadratic complexity with C. Therefore, the total time complexity of the model is $O(C^2)$.

Table 1. Theoretical floating point operations (FLOPs) for spatial attention (SA) and temporal attention (TA) modules. Each module has quadratic complexity with the number of channels. K: convolutional kernel size. C: input and output channels. T: time dimension. H, W: length and width of the feature map.

Module	Step	FLOPs
SA	Simple Mapping	$3C^2 \times H \times W$
	Convolution + Attention	$(K^2 \times C + 3C \times K^2) \times H \times W$
TA	All	$(C^2 + C) \times H \times W \times T + 2C^2 \times T$

3.2. Context Extraction Network

In the video processing task, the original pixel information is gradually lost as the deep learning network continuously encodes and decodes frame sequences [7], and this phenomenon is exacerbated with the increased network depth. To reduce the information loss of the model when processing contextual information, for the output features F_i^{Enc} of each layer of the encoder, we use CE-Net to enhance the feature representation of the encoder, as shown in Figure 3a. CE-Net includes four levels, each processing features from a corresponding layer of the encoder. In each level, we first encode the individual layer F_i^{Enc} with a 1 × 1 convolution, where the output channel dimensions of each level are 4, 8, 16, and 32, respectively. Then, they are upsampled by bilinear interpolation to resize the features to align with the original frames. Next, the features are warped (corresponding to I_0 and I_1) using two DConvs, respectively, thus effectively aggregating contextual features. We concatenate these warped features with the warped input frames and use a Synthesis Network [34] to generate multi-scale intermediate frames $\hat{I}_{0.5}$.



(a) CE-Net

(b) MPFS-Net

(c) SynBlock

Figure 3. The overall structure of the sub-networks. (**a**) Context Extraction Network; (**b**) Multi-scale Prediction Frame Synthesis Network; (**c**) Synthesis Block.

3.3. Multi-Scale Prediction Frame Synthesis Network

Multi-scale frame prediction has been shown to be effective for synthesizing final frames [30]. We designed a synthesis network adapted to our model for predicting frames at different scales, and unlike the method described in [30], MPFS-Net uses only multiple independent synthesis blocks (SynBlocks) to individually predict the feature F_i^{Dec} at different layers of the decoder and at different scales. Note that there are only two input frames in our model. Specifically, the decoder outputs features at three different scale levels, F_0^{Dec}, F_1^{Dec}, and F_2^{Dec}, where F_1^{Dec} and F_2^{Dec} are upsampled and concatenated with F_0^{Dec}, respectively, as inputs to the maximum scale prediction.

As shown in Figure 3c, each SynBlock employs the traditional kernel-based estimation method, which contains one weight estimator, two offset estimators, one occlusion estimator, and two DConvs. SynBlock estimates the parameters of F_i^{Dec}, W_j^i, α_j^i, β_j^i, and M^i. We then apply forward-warping and backward-warping to the different scales of I_0^i and I_1^i with their respective parameters by using DConv, respectively, to obtain P_L^i and P_R^i. Finally, P_L^i and P_R^i are element-wise multiplied with the occlusion map M^i to obtain the i-scale prediction P^i.

The synthesized frames for each scale are obtained from the coarser scale and the current prediction by addition. Firstly, the coarsest scale prediction P^2 is used as the initial value to synthesize the intermediate frame $I''_{0.5}$. Next, the finer scale synthesized frame $I'_{0.5}$ is obtained by up-sampling and combining with the next level of prediction P^1, and so on, until we finally obtain the finest scale synthesized frame $I_{0.5}$.

4. Experiment

4.1. Datasets and Metrics

Our model is trained on a Vimeo90K [8] training dataset. We evaluate our model on various publicly available benchmark datasets, including Vimeo90K [8], Middlebury [9], X4K1000FPS [17], UCF101 [35], SNU-FILM [36], and HD [37]. These benchmark datasets contain rich scenes and large motions, and are widely used for VFI tasks. The details of each dataset are given below:

- **Vimeo90K [8]:** This is a popular dataset widely used in VFI, which consists of three consecutive video frames, the training set contains 51,312 triples with a resolution of 448 × 256. The testing set contains 3782 triples with the same resolution of 448 × 256.
- **Middlebury [9]:** This is a classic visual benchmark for evaluation that provides a wealth of data on realistic scenes. We choose its OTHER testing set, which contains ground-truth and has a resolution of 640 × 480.
- **X4K1000FPS [17]:** This is a 4K video dataset that is typically used to evaluate the performance of models for multi-frame interpolation in ultra-high definition (UHD) resolution scenes. We generate intermediate frames by iteratively using our model to achieve the 8× frame interpolation testing on this dataset at both 4K and 2K resolutions.
- **UCF101 [35]:** This dataset contains rich videos of human behavior and is suitable for video action recognition and video interpolation, among other tasks. It consists of 379 video triples, each with a resolution of 256 × 256.
- **SNU-FILM [36]:** This dataset provides high-quality video sequences and various motion types. It contains 1240 video triplets with a resolution of 1280 × 720. Based on the difficulty of the motion, it is divided into four subsets: easy, medium, hard, and extreme.
- **HD [37]:** The dataset, collected by Bao et al [37], contains 11 videos, including four 1080p, three 720p, and four 1280 × 544 videos. It is often used to evaluate the performance of the model for multi-frame interpolation in high-definition (HD) resolution scenes. We choose videos with resolutions of 1080p and 720p for testing, and generate intermediate frames by iteratively using our model in order to perform the 4× frame interpolation testing on this dataset.
- **Metrics:** We use metrics such as peak signal-to-noise ratio (PSNR), structural similarity (SSIM) [38], and average interpolation error (IE) to evaluate models. On the Middlebury [9] dataset, we calculate its IE value, where a lower IE represents a better performance. Meanwhile, we evaluate the PSNR and SSIM of the models on Vimeo90K [8], X4K1000FPS [17], UCF101 [35], SNU-FILM [36], and HD [37] datasets, where higher PSNR and SSIM indicate better performance.

4.2. Implementation Details

- **Network Architecture:** The encoder has five layers, including an embedding Layer and four PSTAT layers, and the feature channel dimensions of each layer are 32, 32, 64, 128, and 256, respectively, and the downsampling scale factor of each PSTAT layer is 2. There is a skip link between the encoder and decoder. In SA, we set the parameter λ to 0.3 [22], which controls the strength of the convolutional output. We introduce two variants of our model: Ours and Ours-small. Both models are identical in all aspects except for the channel dimensions, where the Ours-small model's channel dimension is set to half of that in the standard Ours model.
- **Training Details:** We crop each training sample of the Vimeo90K [8] training set to 192 × 192 patches and augment the data with random horizontal and vertical flipping as well as time reversal. We use Adan [39] optimizer for end-to-end training, with the hyperparameters β_1, β_2, and β_3 set to 0.98, 0.92, and 0.99, respectively. The training batch size is 8 and the initial learning rate is $2e^{-4}$. We perform 300 epochs using cosine annealing to reduce the learning rate from $2e^{-4}$ to $2e^{-5}$. Our model was trained on an NVIDIA GeForce RTX 3090 (manufactured by NVIDIA Corporation, Santa Clara, CA, USA) with PyTorch 1.12.0, taking about 1 week.

4.3. Comparisons with State-of-the-Art Methods

4.3.1. Quantitative Comparison

We compare our model quantitatively with 17 SOTA methods in the VFI field, including DAIN [5], RIFE [6], RIFE-Large [6], AdaCoF [7], ToFlow [8], ABME [15], XVFI [17], SoftSplat [18], CDFI [19], SepConv [23], EDSC [25], CAIN [36], BMBC [40], IFRNet [41], UPR-Net [42], UPR-Net-large [42], EBME [43]. These methods have been shown to perform significantly well and some have been widely used in industry practice due to their innovation and practicality, such as DAIN [5] and RIFE [6]. Both the SOTA models and our models are trained on the Vimeo90K [8] training set and evaluated on the Vimeo90K [8] testing set, Middlebury [9], UCF101 [35], and SNU-FILM [36]. To comprehensively evaluate the model performance, we not only evaluate the model performance metrics (PSNR, SSIM, and IE) but also compare the parameters and runtimes of the models. This comprehensive evaluation approach is designed to evaluate thoroughly the performance of our models, ensuring that while pursuing high image quality, computational efficiency is also taken into account. This allows for a more accurate assessment of the feasibility and efficacy of the model in real-world applications. For testing the runtime, we tested all the models at 640 × 480 resolution using the same device (NVIDIA GeForce RTX 2080 Ti GPU) and averaged the runtime through 100 iterations.

The results of the quantitative comparison are shown in Table 2. Although our model slightly lags behind the SOTA UPR-Net-large [42] in runtime, our model outperforms it by 0.09 dB on the Vimeo90K [8] testing set, demonstrating a significant performance advantage. Comparing the real-time processing model RIFE-Large [6] and the newly introduced EBME-H* [43], our model outperforms them by 0.27 dB and 0.18 dB, respectively. Furthermore, our model performs excellently in IE on the Middlebury [9], and the quality of the predicted frames is further validated through visual comparisons in Section 4.3.2. On the UCF101 [35], our model also demonstrates the best performance, further proving its robustness across various video scenes. In evaluations on the SNU-FILM [36], particularly in the medium, hard, and extreme subsets, our model consistently achieves the highest PSNR. This outstanding performance demonstrates the effectiveness of our model in handling diverse motion scenes, reflecting the excellent capabilities of our proposed attention mechanism in modeling large motions.

Table 2. Quantitative comparison with state-of-the-art (SOTA) methods. We evaluate Middlebury [9] with the IE, and the other datasets with PSNR/SSIM. The best and second-best results are shown in **red** and blue. M.B. is the abbreviation for Middlebury [9]. "#P" and "#R" represent the number of parameters (in millions) and runtime (in ms), respectively.

Methods	Vimeo90K	UCF101	M.B.	SNU-FILM				#P	#R
				Easy	Medium	Hard	Extreme		
DAIN [5]	34.71/0.9756	34.99/0.9683	2.04	39.73/0.9902	35.46/0.9780	30.17/0.9335	25.09/0.8584	24	151
RIFE [6]	35.61/0.9780	35.28/0.9690	1.96	40.06/0.9907	35.75/0.9789	30.10/0.9330	24.84/0.8534	9.8	**12**
RIFE-Large [6]	36.10/0.9801	35.29/0.9693	1.94	40.02/0.9906	35.92/0.9791	30.49/0.9364	25.24/0.8621	9.8	80
AdaCoF [7]	34.47/0.9730	34.90/0.9680	2.31	39.80/0.9900	35.05/0.9754	29.46/0.9244	24.31/0.8439	22.9	30
ToFlow [8]	33.73/0.9682	34.58/0.9667	2.15	39.08/0.9890	34.39/0.9740	28.44/0.9180	23.39/0.8310	**1.1**	84
ABME [15]	36.18/0.9805	35.38/0.9698	2.01	39.59/0.9901	35.77/0.9789	30.58/0.9364	25.42/**0.8639**	18.1	277
XVFI$_v$ [17]	35.07/0.9681	35.18/0.9519	-	39.78/0.9840	35.37/0.9641	29.91/0.8935	24.73/0.7782	5.5	98
SoftSplat [18]	36.10/0.9700	35.39/0.9520	**1.81**	-	-	-	-	-	-
CDFI [19]	35.17/0.9640	35.21/0.9500	1.98	40.12/0.9906	35.51/0.9778	29.73/0.9277	24.53/0.8476	5	172
SepConv [23]	33.79/0.9702	34.78/0.9669	2.27	39.41/0.9900	34.97/0.9762	29.36/0.9253	24.31/0.8448	21.6	200
EDSC [25]	34.84/0.9750	35.13/0.9680	2.02	40.01/0.9900	35.37/0.9780	29.59/0.9260	24.39/0.8430	8.9	46
CAIN [36]	34.65/0.9730	34.91/0.9690	2.28	39.89/0.9900	35.61/0.9776	29.90/0.9292	24.78/0.8507	42.8	37
BMBC [40]	35.01/0.9764	35.15/0.9689	2.04	39.90/0.9902	35.31/0.9774	29.33/0.9270	23.92/0.8432	11	822
IFRNet [41]	35.80/0.9794	35.29/0.9693	-	40.03/0.9905	35.94/0.9793	30.41/0.9358	25.05/0.8587	5	22
UPR-Net [42]	36.03/0.9801	35.41/0.9698	-	40.37/0.9910	36.16/0.9797	30.67/0.9365	25.49/0.8627	1.7	42
UPR-Net-large [42]	36.28/0.9810	35.43/**0.9700**	-	**40.42/0.9911**	36.24/0.9799	30.81/0.9370	25.58/0.8636	3.7	62
EBME-H* [43]	36.19/0.9807	35.41/0.9697	-	40.28/0.9910	36.07/0.9797	30.64/0.9368	25.40/0.8634	3.9	82
ours-small	36.05/0.9796	35.23/0.9695	1.95	39.91/0.9906	35.87/0.9794	30.68/0.9373	25.46/0.8629	8.7	36
ours	**36.37/0.9811**	**35.44/0.9700**	1.91	40.22/0.9908	**36.25/0.9801**	**30.85/0.9375**	**25.62/0.8639**	17.2	82

To assess the performance of our method in multi-frame interpolation tasks, our model is compared with SOTA methods, such as DAIN [5], RIFE$_m$ [6], ABME [15], IFRNet [41], M2M [44], EMA-VFI-small [45], and EMA-VFI [45]. These SOTA methods are capable of multi-frame interpolation with excellent performance. In this paper, we generate multiple intermediate frames by recursively applying our model to achieve 4× and 8× frame interpolation. Specifically, for the initial input frames I_0 and I_1, we first generate the intermediate frame $I_{0.5}$, followed by using $I_{0.5}$ and I_1, I_0 and $I_{0.5}$ to generate $I_{0.75}$ and $I_{0.25}$, respectively, and so on. As shown in the results in Table 3, our model achieves the second-best performance. Although the performance is slightly below EMA-VFI [45], it is ahead of other SOTA methods and shows satisfactory performance. The results sufficiently prove that our model is able to effectively achieve multi-frame interpolation on datasets with different resolutions.

Table 3. Quantitative comparison with other methods for 4× interpolation on HD [37] and 8× interpolation on XTest [17], evaluated with PSNR. The best and second-best results are shown in **red** and blue.

Method	4×		8×	
	HD (720p)	HD (1080p)	XTest-2K	XTest-4K
DAIN [5]	30.25	-	29.33	26.78
RIFE$_m$ [6]	31.87	34.25	31.43	30.58
ABME [15]	31.43	33.22	30.65	30.16
IFRNet [41]	31.85	33.19	31.53	30.46
M2M [44]	31.94	33.45	32.13	30.88
EMA-VFI-small [45]	32.17	34.65	31.89	30.89
EMA-VFI [45]	**32.38**	**35.28**	**32.85**	**31.46**
Ours	32.21	34.85	32.25	31.09

4.3.2. Qualitative Comparison

In order to evaluate the quality of the video frames generated by our model, besides the quantitative analysis, we also compare the model qualitatively with other SOTA methods, and the results of the visualization comparison on the Vimeo90K [8] testing set are shown in Figure 4. Compared to other methods, our model generates more complete and visually pleasing frames. For example, in the scene (bear's foot) presented in the third row of Figure 4, our method produces less visual distortion. The result in the fourth row (car) presents a clearer tire structure. Moreover, in order to more accurately evaluate the performance of the model in dealing with complex motion, we choose the SNU-FILM [36] for further comparison. The results in Figure 5 indicate that our model generates reasonably clear frames in the scene with rich texture details and fast motion (seabird's wings). In the extreme motion scene (skateboarding), our model generates frames with a more intact structure and clearer details compared to the other three SOTA methods.

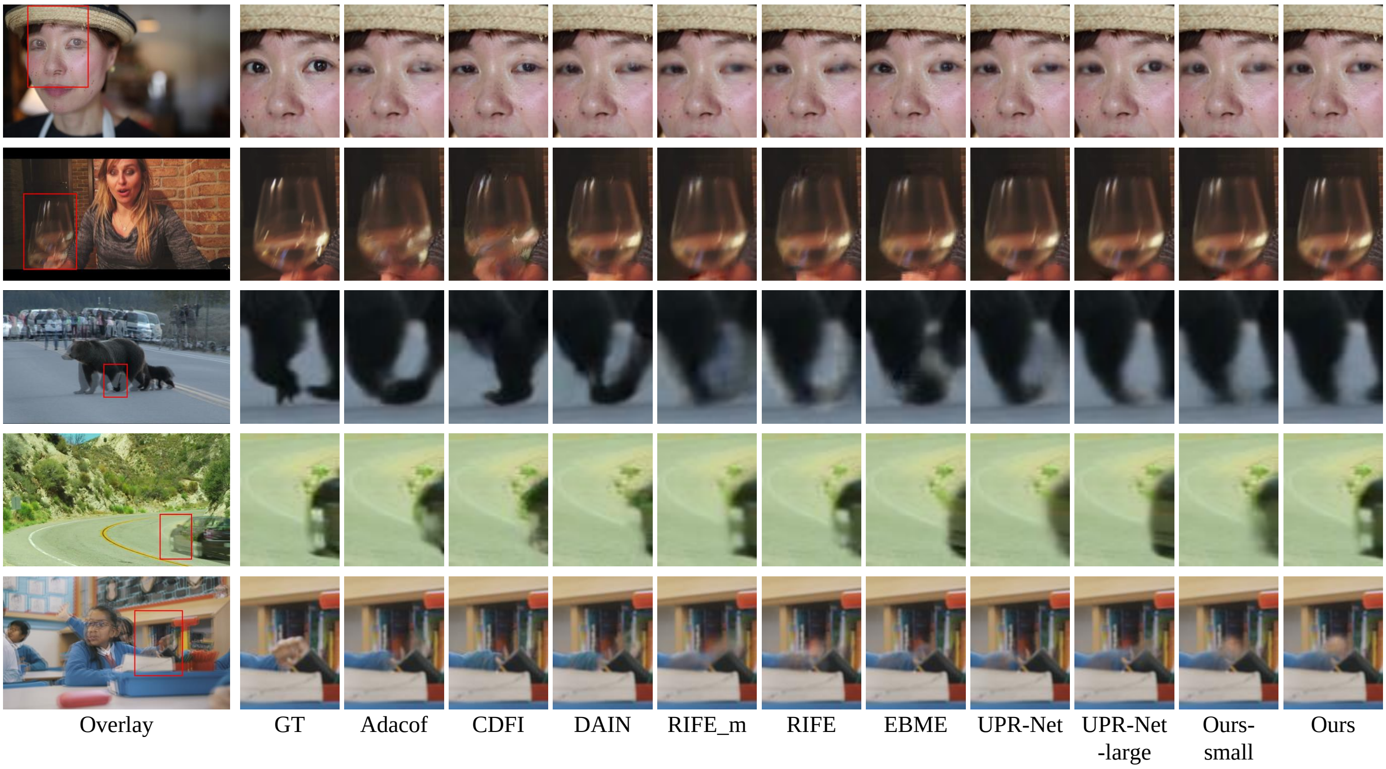


Figure 4. Visual comparison with the state-of-the-art (SOTA) method using the Vimeo90K [8] testing set. The rectangular boxes are the comparison areas. GT is the ground truth.

We choose Vimeo90K [8] and SNU-FILM [36] for visual comparisons because the former includes a rich variety of scenes, while the latter covers situations involving a wide range of motion, both of which contribute to a comprehensive assessment of model performance. Furthermore, the rationale for comparing with different models includes the fact that some models support real-time video processing [6,42], some perform well in quantitative evaluations [19,42], and others represent the latest technological innovations [42,43]. This multi-dimensional comparison not only demonstrates that our model can generate high-quality intermediate frames but also verifies its practical application capability in dealing with various complex scenarios.

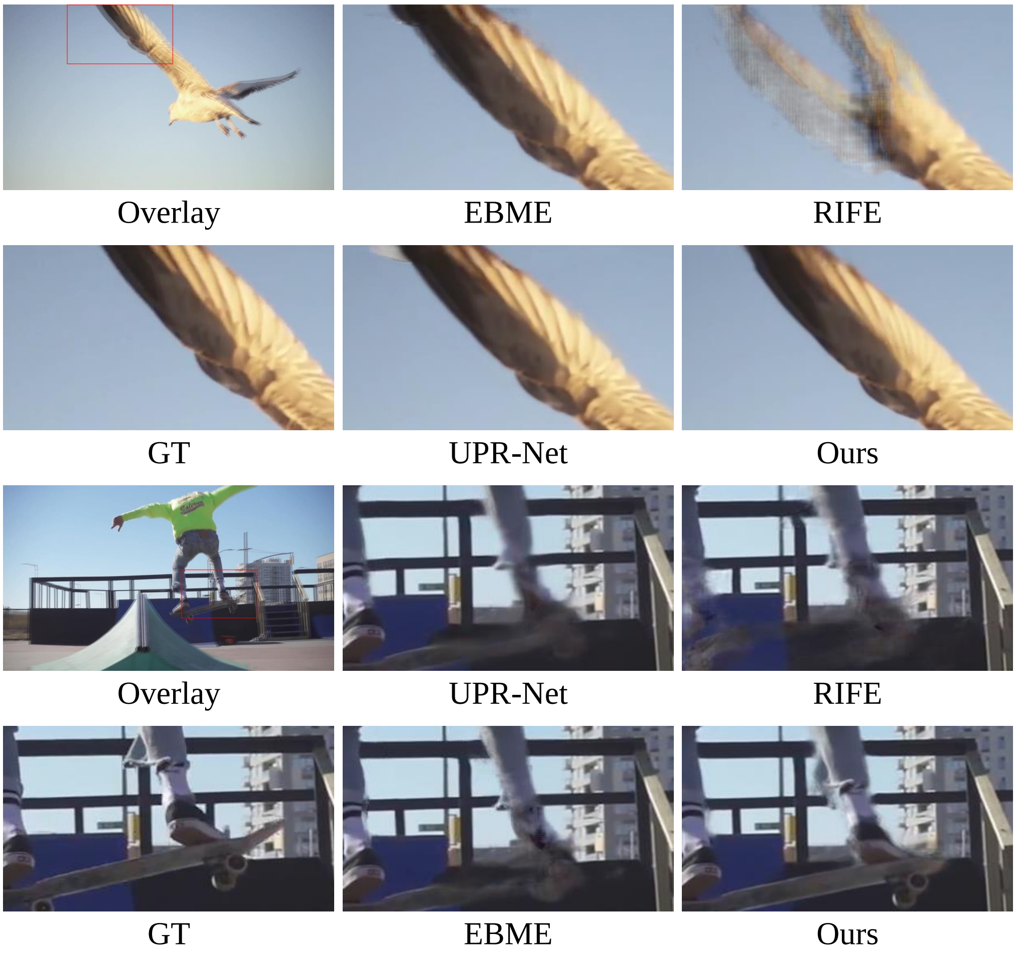


Figure 5. Visualization comparison with other state-of-the-art (SOTA) methods on SNU-FILM [36]. The rectangular boxes are the comparison areas.

4.4. Ablation Study

In this section, we design ablation experiments to assess the impact of specific components on our model's performance. These experiments focus on various aspects such as the overall architecture and layer structure of the model, the PSTAT layer structure, PSTA, and the size of the CE-Net convolutional kernel. The results of these experiments are shown in Tables 4 and 5. We evaluate these configurations on both the Vimeo90K [8] and UCF101 [35] datasets, using the same parameters and training scheme.

- **Ablation Study on Model Layer Structure and Channel Selection.** We investigate the effects of PSTAT's layer structure and initial channel on the model performance, as shown in Table 4. Specifically, we set the number of TRBs in each PSTAT layer to 1 or 2, and each structure corresponds to an initial channel of 16 or 32, respectively. Based on the results in Table 4, we can see that our model structure is scalable and the model performs best when the number of TRBs is 2 and the initial channel dimension is 32. Furthermore, as the number of TRBs decreases, the performance of the model decreases, which indicates the effectiveness of TRBs in the interpolation task.
- **Ablation Study on PSTAT Structure and TRB Structure.** In the PSTAT layer, in order to more thoroughly verify the effectiveness and scalability of PSTA, we use a regular convolutional layer instead of PSTA, and the results are shown in Table 5. In the same number of TRBs, the PSNR for the TRB with regular convolution is consistently lower than for the TRB with PSTA. When comparing the results of TRB = 2 with convolution to TRB = 1 with one PSTA block, it was found that the PSNR of the former was lower

than the latter. These results indicate that PSTA is superior to convolution and that PSTA is more suitable for modeling inter-frame motion. Additionally, as the number of PSTA blocks increases, model performance improves, further demonstrating our model's scalability.

- **Ablation Study on PSTA.** We perform an ablation study on the PSTA structural design in order to analyze the effect of SA and TA on the model performance, and the results are shown in Table 5. We first evaluate the performance impact of SA and TA on the model. In particular, the PSNR of the model is 35.61 dB when we use a convolutional layer instead of SA, and the PSNR of the model without TA is 36.19 dB. This result shows that the performance of the model degrades when either SA or TA is missing, and SA has a greater impact on our model. This indicates that our proposed PSTA is very effective and it allows our model to aggregate both inter-frame and intra-frame information without increasing the computational overhead. It also shows the superiority of the parallel mechanism. In addition, we remove the self-attention mechanism and convolution from the SA, respectively, and the results show that both mechanisms affect the performance of the model, and the lack of the self-attention mechanism has a greater impact on the model performance than the lack of convolution. This result suggests that the self-attention mechanism is more suitable than convolution for modeling large motions for the VFI task, and indirectly shows that the Transformer-based model proposed in this paper outperforms the CNN-based model.
- **Ablation Study on Model Architecture Design.** For CE-Net and MPFS-Net, we conduct a simple comparison experiment, and we construct three model structures, namely: the model without CE-Net, the model without MPFS-Net, and the model without CE-Net and MPFS-Net. As shown by the results in Table 5, when the model lacks CE-Net and MPFS-Net, the model performs poorly, especially when both are missing. These results show that CE-Net and MPFS-Net are beneficial for our model and can fully realize the performance of the Transformer-based structure. They also enable our model to learn multi-scale information and synthesize high-quality video frames.
- **Ablation Study on Conv Scheme in CE-Net.** To investigate the strategy of using the 1×1 convolution in CE-Net, we use a simple model structure (1 TRB + 1 PSTA block). We then replace the convolution kernel with a 3×3 kernel and evaluate the model for CE-Net with different sizes of convolution kernels. Comparison of the results in the last two rows of Table 5 show that the large-size convolutional kernel performs poorly in our model and fails to focus on more details. In contrast, the 1×1 convolution can aggregate more contextual information and is more suitable for CE-Net.
- **Visual Ablation Study.** Besides quantitative comparisons of ablation studies, we also perform qualitative comparisons of PSTA, CE-Net, and MPFS-Net on the Middlebury [9], as shown in Figure 6. In particular, the model without SA generates significantly blurrier intermediate frames. In the second row, all models generate tennis balls with sharp edges, which indicates that the Transformer structure is able to robustly model the similarity of long-range pixels for objects with regular shapes. Furthermore, the full model is able to generate sharper and more detailed intermediate frames compared to the version without key modules. This comparison clearly shows the important contribution of each module in improving the quality of frame synthesis. However, in the second row, there is still improvement in the performance of all the models for human body motions, especially finger joints, which will be the focus of our future research.

Table 4. Ablation studies for transformer residual blocks and channels. "X-X-X-X" indicates the number of Transformer residual blocks in each corresponding layer.

Architecture	Channel	Vimeo90K		UCF101	
		PSNR	SSIM	PSNR	SSIM
1-1-1-1	16	35.46	0.9774	34.60	0.9626
2-2-2-2	16	36.05	0.9796	35.23	0.9695
1-1-1-1	32	36.11	0.9801	35.31	0.9698
2-2-2-2	32	36.37	0.9811	35.44	0.9700

Table 5. Ablation studies of encoder layer structure, parallel spatio-temporal attention (PSTA), model architecture, and the size of the CE-Net convolutional kernel. "w/" denotes "with" and "w/o" denotes "without".

Setting	Vimeo90K		UCF101	
	PSNR	SSIM	PSNR	SSIM
PSTAT Layer Structure Design				
1 TRB w/Conv 3 × 3	34.21	0.9723	34.07	0.9527
1 TRB w/1 PSTA block	35.59	0.9794	34.85	0.9633
1 TRB w/2 PSTA blocks	36.11	0.9801	35.31	0.9698
2 TRB w/Conv 3 × 3	35.34	0.9692	34.41	0.9628
2 TRB w/1 PSTA block	36.30	0.9811	35.34	0.9699
2 TRB w/2 PSTA blocks	36.37	0.9811	35.44	0.9700
PSTA Design				
Ours w/o SA	35.61	0.9779	34.90	0.9635
Ours w/o TA	36.19	0.9802	35.32	0.9698
SA w/o Attention	35.64	0.9782	35.05	0.9686
SA w/o Conv	35.94	0.9801	35.26	0.9691
Model Architecture Design				
Ours w/o MPFS-Net + CE-Net	35.58	0.9757	34.82	0.9633
Ours w/o CE-Net	36.12	0.9799	35.33	0.9698
Ours w/o MPFS-Net	35.93	0.9791	35.24	0.9690
CE-Net Conv Scheme (Model w/ 1 TRB + 1 PSTA block)				
CE-Net - Conv 3 × 3	35.52	0.9792	34.68	0.9629
CE-Net - Conv 1 × 1	35.59	0.9794	34.85	0.9633

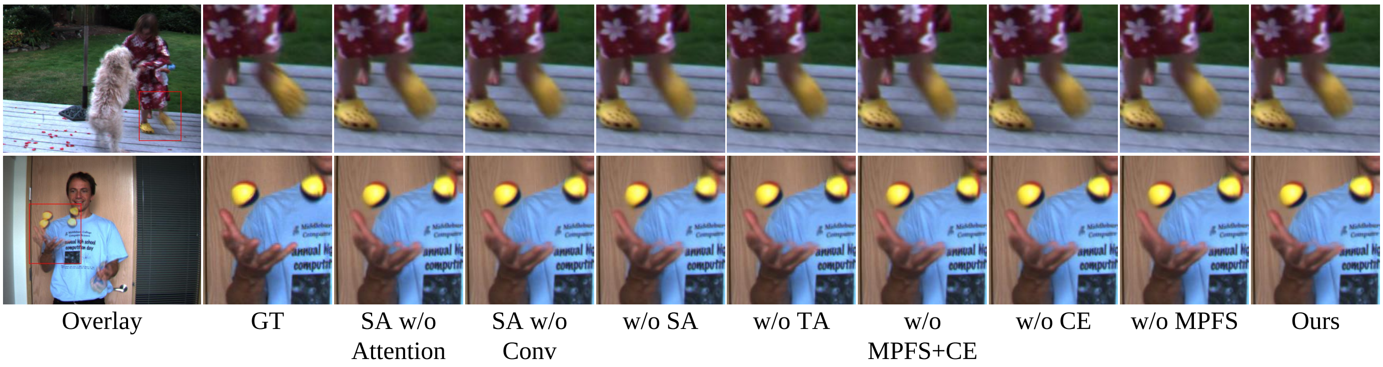


Figure 6. Visualization comparison of ablation studies. The testing dataset is Middlebury [9]. "w/o" denotes "without". The rectangular boxes are the comparison areas.

5. Limitations and Future Work

Although our approach has achieved significant results, there are still some limitations that need to be further investigated. Currently, our model only generates intermediate frames and is limited to accepting only two consecutive frames as input, which means that the information in multiple consecutive frames cannot be fully utilized. In future work, we aim to develop a model that accepts multiple frame inputs and extend our method to handle frame interpolation at arbitrary time steps.

6. Conclusions

In this study, we propose a new model using Transformer architecture for VFI. The model contains the parallel spatio-temporal attention mechanism for extracting inter-frame and intra-frame motion information and modeling long-range pixel dependencies. Particularly, it is worth mentioning that our proposed parallel spatio-temporal attention mechanism, based on a simple structure, facilitates the interaction of motion information across the time dimension. It effectively avoids the additional computational overhead typically associated with reusing attention mechanisms. Extensive experimental results show that our model demonstrates excellent performance on multiple standard datasets and is able to generate more visually pleasing intermediate frames compared to existing methods.

Author Contributions: Conceptualization: X.N., F.C. and Y.D.; data curation: X.N., Y.L. and F.C.; formal analysis: X.N.; investigation: X.N., F.C. and Y.L.; methodology: X.N.; resources: Y.D.; software: X.N.; validation: X.N., F.C. and Y.D.; visualization: X.N. and F.C.; writing—original draft: X.N.; writing—review and editing: F.C., Y.L. and Y.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Nos. 61303093 and 61402278) and the Shanghai Natural Science Foundation (No. 19ZR1419100).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this study are all derived from public domain resources and have been detailed in the reference section. For specific details and citations, please refer to the list of references.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wu, C.Y.; Singhal, N.; Krahenbuhl, P. Video compression through image interpolation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 416–431.
2. Kim, S.Y.; Oh, J.; Kim, M. Fisr: Deep joint frame interpolation and super-resolution with a multi-scale temporal loss. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 11278–11286.
3. Haris, M.; Shakhnarovich, G.; Ukita, N. Space-time-aware multi-resolution video enhancement. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2859–2868.
4. Jiang, H.; Sun, D.; Jampani, V.; Yang, M.H.; Learned-Miller, E.; Kautz, J. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9000–9008.
5. Bao, W.; Lai, W.S.; Ma, C.; Zhang, X.; Gao, Z.; Yang, M.H. Depth-aware video frame interpolation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3703–3712.
6. Huang, Z.; Zhang, T.; Heng, W.; Shi, B.; Zhou, S. Real-time intermediate flow estimation for video frame interpolation. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022; pp. 624–642.
7. Lee, H.; Kim, T.; Chung, T.Y.; Pak, D.; Ban, Y.; Lee, S. AdaCoF: Adaptive collaboration of flows for video frame interpolation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 5315–5324.
8. Xue, T.; Chen, B.; Wu, J.; Wei, D.; Freeman, W. Video enhancement with task-oriented flow. *Int. J. Comput. Vis.* **2018**, *127*, 1106–1125. [CrossRef]
9. Baker, S.; Scharstein, D.; Lewis, J.; Roth, S.; Black, M.J.; Szeliski, R. A database and evaluation methodology for optical flow. *Int. J. Comput. Vis.* **2007**, *92*, 1–31. [CrossRef]
10. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 10012–10022.

11. Im, S.K.; Chan, K.H. Distributed Spatial Transformer for Object Tracking in Multi-Camera. In Proceedings of the 2023 25th International Conference on Advanced Communication Technology (ICACT), Pyeongchang, Republic of Korea, 19–22 February 2023; pp. 122–125.
12. Thawakar, O.; Narayan, S.; Cao, J.; Cholakkal, H.; Anwer, R.M.; Khan, M.H.; Khan, S.; Felsberg, M.; Khan, F.S. Video instance segmentation via multi-scale spatio-temporal split attention transformer. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022; pp. 666–681.
13. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 213–229.
14. Lu, L.; Wu, R.; Lin, H.; Lu, J.; Jia, J. Video frame interpolation with transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 3532–3542.
15. Park, J.; Lee, C.; Kim, C.S. Asymmetric bilateral motion estimation for video frame interpolation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 10–25 June 2021; pp. 14539–14548.
16. Kalluri, T.; Pathak, D.; Chandraker, M.; Tran, D. Flavr: Flow-agnostic video representations for fast frame interpolation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Vancouver, BC, Canada, 18–22 June 2023; pp. 2071–2082.
17. Sim, H.; Oh, J.; Kim, M. Xvfi: Extreme video frame interpolation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 14489–14498.
18. Niklaus, S.; Liu, F. Softmax splatting for video frame interpolation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 5437–5446.
19. Ding, T.; Liang, L.; Zhu, Z.; Zharkov, I. Cdfi: Compression-driven network design for frame interpolation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 8001–8011.
20. Zhang, D.; Huang, P.; Ding, X.; Li, F.; Zhu, W.; Song, Y.; Yang, G. L2BEC2: Local Lightweight Bidirectional Encoding and Channel Attention Cascade for Video Frame Interpolation. *ACM Trans. Multimed. Comput. Commun. Appl.* **2023**, *19*, 1–19.
21. Ding, X.; Huang, P.; Zhang, D.; Liang, W.; Li, F.; Yang, G.; Liao, X.; Li, Y. MSEConv: A Unified Warping Framework for Video Frame Interpolation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2024**. [CrossRef]
22. Ning, X.; Li, Y.; Feng, Z.; Liu, J.; Ding, Y. An Efficient Multi-Scale Attention Feature Fusion Network for 4k Video Frame Interpolation. *Electronics* **2024**, *13*, 1037. [CrossRef]
23. Niklaus, S.; Mai, L.; Liu, F. Video frame interpolation via adaptive separable convolution. In Proceedings of the IEEE international conference on computer vision, Venice, Italy, 22–29 October 2017; pp. 261–270.
24. Cheng, X.; Chen, Z. Video frame interpolation via deformable separable convolution. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 10607–10614.
25. Cheng, X.; Chen, Z. Multiple video frame interpolation via enhanced deformable separable convolution. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 7029–7045. [CrossRef] [PubMed]
26. Im, S.K.; Chan, K.H. Local feature-based video captioning with multiple classifier and CARU-attention. *IET Image Proc.* 2024, *early view*. [CrossRef]
27. Zhu, X.; Hu, H.; Lin, S.; Dai, J. Deformable convnets v2: More deformable, better results. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9308–9316.
28. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
29. Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Van Gool, L.; Timofte, R. SwinIR: Image restoration using Swin Transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 1833–1844.
30. Shi, Z.; Xu, X.; Liu, X.; Chen, J.; Yang, M.H. Video frame interpolation transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 17482–17491.
31. Hendrycks, D.; Gimpel, K. Gaussian error linear units (gelus). *arXiv* **2016**, arXiv:1606.08415.
32. Lu, Z.; Li, J.; Liu, H.; Huang, C.; Zhang, L.; Zeng, T. Transformer for single image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 457–466.
33. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018, pp. 7132–7141.
34. Fourure, D.; Emonet, R.; Fromont, E.; Muselet, D.; Tremeau, A.; Wolf, C. Residual conv-deconv grid network for semantic segmentation. *arXiv* **2017**, arXiv:1707.07958.
35. Soomro, K.; Zamir, A.R.; Shah, M. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv* **2012**, arXiv:1212.0402.
36. Choi, M.; Kim, H.; Han, B.; Xu, N.; Lee, K.M. Channel attention is all you need for video frame interpolation. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 10663–10671.
37. Bao, W.; Lai, W.S.; Zhang, X.; Gao, Z.; Yang, M.H. Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 933–948. [CrossRef] [PubMed]
38. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]

39. Xie, X.; Zhou, P.; Li, H.; Lin, Z.; Yan, S. Adan: Adaptive Nesterov Momentum Algorithm for Faster Optimizing Deep Models. *arXiv* **2022**, arXiv:2208.06677.
40. Park, J.; Ko, K.; Lee, C.; Kim, C.S. Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 109–125.
41. Kong, L.; Jiang, B.; Luo, D.; Chu, W.; Huang, X.; Tai, Y.; Wang, C.; Yang, J. Ifrnet: Intermediate feature refine network for efficient frame interpolation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 1969–1978.
42. Jin, X.; Wu, L.; Chen, J.; Chen, Y.; Koo, J.; Hahm, C.h. A unified pyramid recurrent network for video frame interpolation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 1578–1587.
43. Jin, X.; Wu, L.; Shen, G.; Chen, Y.; Chen, J.; Koo, J.; Hahm, C.h. Enhanced bi-directional motion estimation for video frame interpolation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 5049–5057.
44. Hu, P.; Niklaus, S.; Sclaroff, S.; Saenko, K. Many-to-many splatting for efficient video frame interpolation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 3553–3562.
45. Zhang, G.; Zhu, Y.; Wang, H.; Chen, Y.; Wu, G.; Wang, L. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 5682–5692.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.