ConvTransformer: A Convolutional Transformer Network for Video Frame Synthesis

Zhouyong Liu Shun Luo Wubin Li Jingben Lu Yufan Wu Chunguo Li
Luxi Yang
School of Information Science and Engineering, Southeast University

DVF 23.19/0.80 | MCNet 27.73/0.93 | Ours 33.07/0.97 | Ground truth PSNR/SSIM

Figure 1. **Example of video frame extrapolation.** Top is the extrapolated result, middle is the zoomed local details and bottom is the occlusion map computed with ground truth.

Abstract

Deep Convolutional Neural Networks (CNNs) are powerful models that have achieved excellent performance on difficult computer vision tasks. Although CNNS perform well whenever large labeled training samples are available, they work badly on video frame synthesis due to objects deforming and moving, scene lighting changes, and cameras moving in video sequence. In this paper, we present a novel and general end-to-end architecture, called ***convolutional Transformer*** *or* ***ConvTransformer****, for video frame sequence learning and video frame synthesis. The core ingredient of ConvTransformer is the proposed attention layer, i.e., multi-head convolutional self-attention, that learns the sequential dependence of video sequence. Our method ConvTransformer uses an encoder, built upon multi-head convolutional self-attention layers, to map the input sequence to a feature map sequence, and then another deep networks, incorporating multi-head convolutional self-attention layers, decode the target synthesized frames from the feature maps sequence. Experiments on video future frame extrapolation task show ConvTransformer to be superior in quality while being more parallelizable to recent approaches built upon convoltuional LSTM (ConvLSTM). To the best of our knowledge, this is the first time that ConvTransformer architecture is proposed and applied to video frame synthesis.*

1. Introduction

Video frame synthesis, aiming to synthesize spatially and temporally coherent intermediated frames between two consecutive reals frames or synthesize the future frames of a frame sequence, is a classical and fundamental problem in video processing and computer vision community. The abrupt motion artifacts and temporal aliasing in video sequence can be compressed with the help of video frame synthesis, and hence it can be applied to numerous applications ranging from motion deblurring [5] to video frame rate up-sampling [7, 3], video editing [23, 37], novel view synthesis [9] and autonomous vehicle [34].

The traditional video frame synthesis pipeline usually involves two consecutive steps, i.e., optical flow estimation and optical based frame interpolation or extrapolation [32, 36]. Although these methods perform well when optical flow is accurately estimated, they would generate motion blur and artifacts with inaccurate optical estimation.

Deep neural networks based approaches have been successfully applied to numerous computer vision tasks, such as classification [13], segmentation [24] and visual tracking [15], and promote the development of video frame interpolation and extrapolation. Niklaus *et al.* considered frame interpolation as a local convolution over the two origin frames and used a convolutional neural network (CNN) to learn a spatially-adaptive convolutional kernel for each pixel [20, 21]. Although their methods achieves high-quality results, these methods suffer from heavy computation and are also sensitive to motion. Recently, with the success of applying deep neural networks to motion estimation, deep warping models consisting of motions estimation sub-network and motion compensation sub-network have been proposed for video frame interpolation and gains significant augmentation on performance [12, 2, 22, 1]. Unlike the video frame interpolation task, video frame extrapolation can be treated as a prediction problem, i.e., the future frame is predicted by the previous video frame sequence. Liu *et al.* exploits the voxel flow for predicting future frames and achieves relatively high-quality results [17]. Lately, Villegas *et al.* considered frame extrapolation as a sequence prediction problem, and proposed a prediction networks MCNet [30] which is built upon the convolutional Long-Short-Term Memory (ConvLSTM) [25]. However, the recurrent model of ConvLSTM makes MCNet hard to train, and meanwhile suffers from heavy computational burden when the length of sequence is large.

Although the deep warping based interpolation models obtain better results, these methods are mainly developed on two consecutive frames for frame interpolation, while the high-order motion information of video frame sequence is ignored and not be well exploited. Besides, the ConvLSTM based models not only suffer from heavy computation due to the recurrent architecture, but also face the hard training problem. Last but not least, the deep warping models cannot be directly used for video frame extrapolation task without any modification, and the ConvLSTM based extrapolation algorithms will suffer from heavily performance degradation as compared with deep warping models because of the later frames is unused.

In order to bridge this gap, we, in this work, propose a general end-to-end video frame synthesis network, i.e., convolutional Transformer (ConvTransformer), which simplifies the video frame synthesis as an encoder and decoder problem. Through the proposed multi-head convolutional self-attention mechanism, ConvTransformer extracts the high-order motion information existing in video sequence, and exploits it for synthesizing the target interpolated frames. Unlike the recurrent operation of ConvLSTM based approaches, ConvTransformer can be implemented in parallel both in training and testing stage. Different from the standard Transformer used in nature language processing (NLP) community, the standard operation in ConvTransformer is 2D convolutional operation and the attention operation is conducted on feature maps.

We evaluate ConvTransformer on several benchmarks, i.e. UCF101 [26], Vimeo90K [35], Sintel [11], REDS [19], HMDB [14] and Adobe240fps [28], against several competitive interpolation and extrapolation baselines. Experiments show that our ConvTransformer achieves comparable results against state-of-the-art methods.

The main contributions of this paper are therefore as follows.

- A novel architecture named ConvTransformer is proposed for video frame synthesis.
- A novel attention assigned multi-head convolutional self-attention is proposed for modeling the long-range spatial and temporal dependence on video frame sequence.
- The effectiveness and superiority of the proposed ConvTransformer have been comprehensively analyzed in this paper.

2. Related Works

2.1. Video Frame Synthesis

Video frame synthesis is a longstanding low-level computer vision problem, which is very challenging and inherently ill-posed since the multi-modal distribution of natural images and videos. Numerous solutions, over the past decades, have been proposed for this problem, and achieves substantial improvement. Traditional methods usually exploit the optical flow of two consecutive frames or frame sequence, and then synthesize the target frames guided by the optical flow map. These methods, however, are sensitive to the motion and brightness changing in videos.

Instead of relying on optical flow, several deep neural networks based algorithms were proposed. Specifically, Long *et al.* train a generic CNN to directly synthesize the middle frame [18], which sometimes introduces motion blurriness. The deep voxel flow networks [17], incorporating 3D optical flow across space and time, is proposed to model the optical flow between two original frames and then warp input frames based on a trilinear sampling. Although the synthesized frames suffer from less motion blurry with the help of flow, the flow estimation is still challenging for large motion scenes and the inaccurate estimated flow always result in distorted middle frames. Niklaus *et al.* consider the frame interpolation as a local convolution over the two origin frames and use a convolutional neural networks (CNN) to learn a spatially-adaptive convolutional kernel for each pixel [20, 21]. Although their methods achieves high-quality results, these methods suffer from heavy computation and are sensitive to motion.

Recently, with the success of applying deep neural networks for motion estimation, deep models consisting of motion estimation sub-network and motion compensation

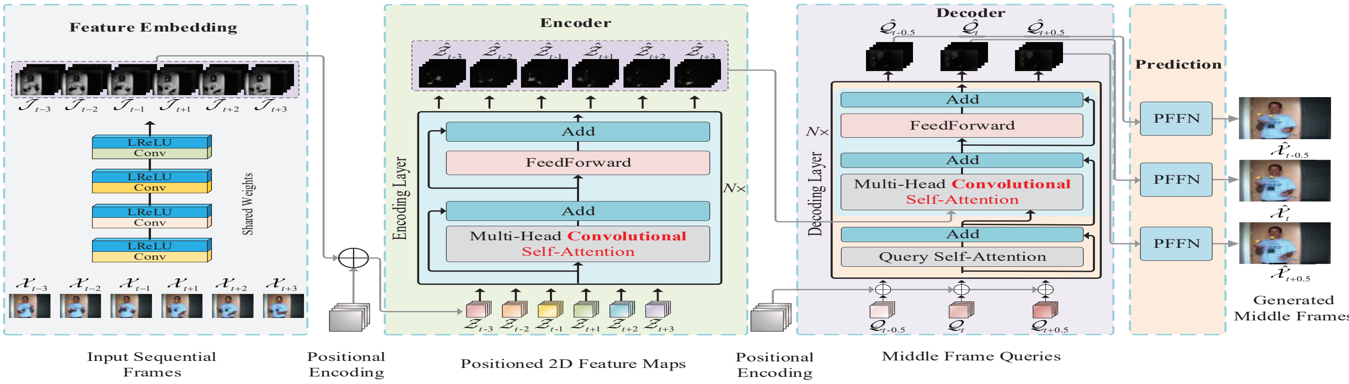

Figure 2. The architecture of the proposed ConvTransformer $\mathcal{G}_{\theta_G}$

sub-network have been proposed for video frame interpolation [12, 2, 22, 1]. Although these methods obtain not only good interpolation results but also promising unsupervised flow estimation results, these warping based deep models are mainly developed based on two consecutive frames for frame interpolation, while the higher-order motion information of video frame sequence is ignored and not be well exploited. Unlike the warping based methods used in frame interpolation, Vilegas *et al.* modeled the video frame extrapolation problem as a recurrent frames prediction task and proposed a ConvLSTM based method MCNet for extrapolation. However, MCNet is hard to establish efficient relation of two frames with large distance, and it suffer from high complexity in computation due to the recurrent model. Besides, these extrapolation methods performs bad as compared with that optical based interpolation methods[12, 2, 22, 1]. In contrast, we propose a novel and general architecture ConvTransformer to model the video frame synthesis task, including extrapolation and interpolation, as a sequence prediction problem. we propose a multi-heal convolutional self-attention mechanism to model the long-term dependence between two frames, and the distance of each paired frames are 1. Given a n-gram sequence, the synthesized middle frames not only relate to the adjacent two frames as implemented in optical based methods, but also have a long-term dependence on further and later video frames.

2.2. Transformer Network

Transformer [29] is a novel architecture for learning long-range sequential dependency, which abandons the traditional building style of directly using RNN or LSTM architecture. It has been successfully applied to numerous natural language processing (NLP) tasks, such as machine translation and speech recognition. Based on Transformer, a series of modifications have been proposed and extensively investigated. Specifically, Zihang Dai *et al.* concentrate on removing the encoding length limitation of Transformer. They exploit the self-attention layer used in last segment to build the connection between segments [8]. Also, Relative Positional Encodings are included to suit their network. Considering the large computation of transformer, Wu et al. reduce the channel number of Transformer. They think that traditional attention has many redundant data, and they introduce Long-Short Range Attention (LARA) to lower the memory cost [33]. Also focusing on the computation cost, Iz Beltagy et al. point out that as the input length increase, the number of parameters computed in self-attention layer grows quadratically [4]. They introduce sparse attention using different sliding windows, all of which are linear to input length.

Recently, Nicolas *et al.* expand the basic Transformer architecture to the field of object detection and propose the DETR algorithm [6], which achieves competitive result on COCO dataset as compared with Faster RCNN. Through collapsing the spatial dimensions (two dimension) into one dimension, DETR reasons about the relations of pixels and the global image context. Although DETR has successfully applied Transformer for computer vision task object detection, it is hard use the basic Transformer to model the long term dependence among the two dimensional video frames, which not only temporally related, but also spatially related. In order to overcome this issue, a convolutional Transformer (ConvTransformer) is proposed in this work, and has been successfully applied to video frame synthesis. The experiment results show that the simplified architecture ConvTransformer achieves competitive results as compared to state-of-the-art well-designed networks, such as MCNet, DAIN and BMBC. To the best of our knowledge, it is the first time that ConvTransformer architecture is proposed.

3. Convolutional Transformer Architecture

The overall ConvTransformer networks $\mathcal{G}_{\theta_G}$, as illustrated in Figure 2, are five main components, that is, feature embedding module $\mathcal{F}_{\theta_F}$, positional encoding module $\mathcal{P}_{\theta_P}$, encoder module $\mathcal{E}_{\theta_E}$, query decoder module $\mathcal{D}_{\theta_D}$, and syn-

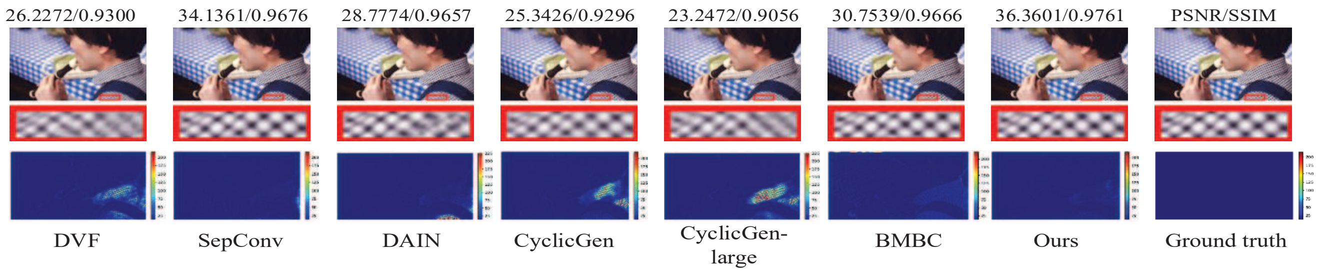

Figure 3. Visual Comparisons of ConvTransformer with other state-of-the-art video frame interpolation methods: DVF [17], SepConv [21], DAIN [1], CyclicGen [16] and BMBC [22]

thesis feed-forward networks $\mathcal{S}_{\theta_S}$. In this section, we first provide an overview of video frame synthesis pipeline realised by ConvTransformer architecture, and then make an illustrated introduction of the proposed ConvTransformer. Finally, the implementation details and training loss is introduced.

3.1. Algorithm Review

Given a video frame sequence $\tilde{\mathcal{X}}= \{\tilde{\mathcal{X}}_0,\tilde{\mathcal{X}}_1, \cdots ,\tilde{\mathcal{X}}_n\} \in \mathbb{R}^{H\times W\times C}$, where n is the length of sequence and H, W and C denote height, width, and number of channels, respectively, our goal is to synthesize intermediate frames $\hat{\mathcal{X}} = \{\hat{\mathcal{X}}_{i+t_0}, \hat{\mathcal{X}}_{i+t_1}, \cdots, \hat{\mathcal{X}}_{i+t_k}\}$ at time $t_k \in [0, 1]$, or extrapolate future frames $\hat{\mathcal{X}} = \{\hat{\mathcal{X}}_{n+t_0}, \hat{\mathcal{X}}_{n+t_1}, \cdots, \hat{\mathcal{X}}_{n+t_k}\}$ at time $t'_m \in \mathbb{N}$. The future frame forecasting problem can be viewed as:

$$\hat{\mathcal{X}}_{n+t_0}, \cdots, \hat{\mathcal{X}}_{n+t_k} = \underset{\mathcal{X}_{n+t_0}, \cdots, \mathcal{X}_{n+t_k}}{\text{argmax}} \; p(\mathcal{X}_{t+1}, \cdots, \mathcal{X}_{t+K}|\tilde{\mathcal{X}}) \tag{1}$$

Firstly, the **Feature Embedding** module embeds the input video frames, and then generates representative feature maps. Subsequently, the extracted feature maps of each frame are added with the positional map, which are used for positional identity. Next, the positioned frame feature maps are passed as inputs to the **Encoder** to exploit the long-range sequential dependence among each frame in video sequence. After getting the encoded high-level feature maps, the high-level feature maps and positioned frame queries are simultaneously passed into the **Decoder**, and then the sequential dependence between the query frames and input sequential video frames are decoded. Finally, the decoded feature maps are fed into the **Synthesis Feed-Forward Networks (SFFN)** to generate the final middle interpolated frames or extrapolated frames.

3.2. Feature Embedding $\mathcal{F}_{\theta_F}$

In order to extract a compact feature representation for subsequent effective learning, a representative feature is computed by a 4-layer convolution with Leaky ReLu activation function and hidden dimension d_{model}. Given a video frame $\mathcal{X}_i \in \mathbb{R}^{H\times W\times 3}$, the embedded feature maps $\mathcal{H}_i \in \mathbb{R}^{H\times W\times d_{model}}$ can be presented with the following equation:

$$\mathcal{J}_i = \mathcal{F}_{\theta_F}(\mathcal{X}_i), \quad i \in [1, n] \tag{2}$$

It is worth mentioning that all input video frames not only share the same embedding net architecture $\mathcal{F}_{\theta_F}$, but also the parameters θ_F.

3.3. Positional Encoding $\mathcal{P}_{\theta_P}$

Since our model contains no recurrence across the video frame sequence, in order for the model to utilize the order of the video frame sequence, some information about relative or absolute position of the video frames must be injected in the frames' feature map. To this end, "positional encodings" are added at each layer in encoder and decoder. It is noted that the positional encodings in ConvTransformer is a 3D tensor which is different from that in original Transformer architecture build for vector sequence. The positional encodings have the same dimension as the frame feature maps, so that the two can be summed. In this work, we use sine and cosine functions of different frequencies:

$$\text{Pos\_Map}_{(p,(i,j,2k))} = \sin(n/10000^{2k/d_{model}}) \tag{3}$$

$$\text{Pos\_Map}_{(p,(i,j,2k+1))} = \cos(n/10000^{2k/d_{model}}) \tag{4}$$

where p is the position token, (i, j) represent the spatial location of features and the channel dimension is represented as $2k$. That is, each dimension of the positional encoding corresponds to a sinusoid. The wavelengths form a geometric progression from 2π to $10000 * 2\pi$. We choose this function because it would allow the model to easily learn to attend by relative positions since for any fixed offset m, $\text{Pos\_Map}_{(p+m)}$ can be represented as a linear function of $\text{Pos\_Map}_{(p)}$.

Given an embedded feature maps $\mathcal{J}_i$, the positioned embedding can be viewed as the following equation:

$$\mathcal{Z}_i = \mathcal{J}_i \oplus \text{Pos\_Map}_{(i)}, \quad i \in [1, n] \tag{5}$$

where the $\oplus$ operation represents element-wise tensor addition.

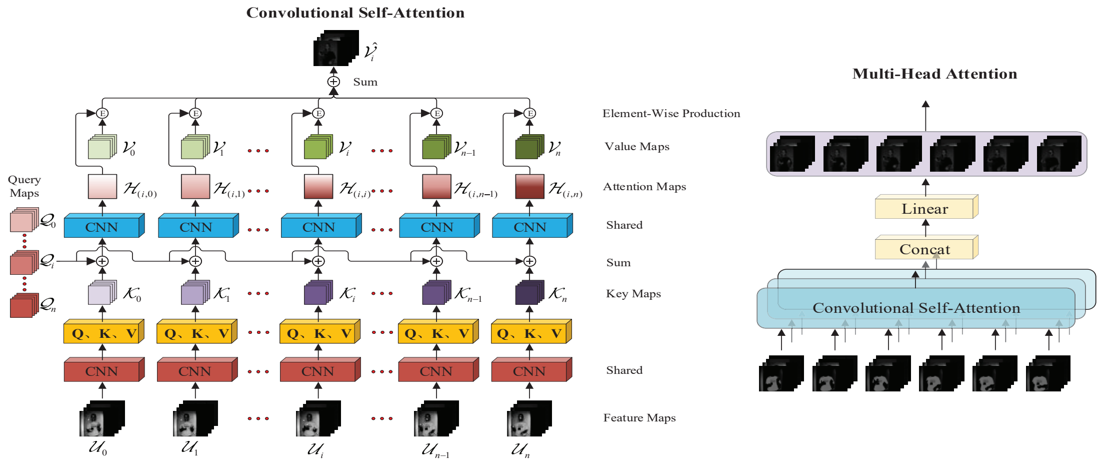

Figure 4. (left) Convolutional Self-Attention. (right) Multi-Head Attention in parallel.

3.4. Encoder $\mathcal{E}_{\theta_E}$ and Decoder $\mathcal{D}_{\theta_D}$

Encoder: As shown in Figure 2, the encoder is modeled as a stack of N identical layers consisting of two sub-layers, i.e., multi-head convolutional self-attention layer and a simple 2D convolutional feed-forward networks. The residual connection is adopted around each of the two sub-layers. To facilitate these residual connections, all sub-layers in the model, as well as the embedding layers, produce outputs of the same dimensional $d_{model} = 32$. Given a positioned feature sequence $\mathcal{Z}= \{\mathcal{Z}_0,\mathcal{Z}_1, \cdots ,\mathcal{Z}_i, \cdots ,\mathcal{Z}_{n-1},\mathcal{Z}_n\} \in \mathbb{R}^{H\times W\times d}$, the identified feature sequence $\hat{\mathcal{Z}}= \{\hat{\mathcal{Z}}_0,\hat{\mathcal{Z}}_1, \cdots ,\mathcal{Z}_n\} \in \mathbb{R}^{H\times W\times d}$ can be learned, and the encoding operation can be represented as:

$$\hat{\mathcal{Z}} = \mathcal{E}_{\theta_E}(\mathcal{Z}) \tag{6}$$

Decoder: The decoder is also composed of a stack of N identical layers, which consists of three sub-layers. In addition to the same two sub-layers as implemented in Encoder, an additional layer called query self-attention is inserted to perform the convolutional self-attention over the output frame queries. Given a query sequence $\mathcal{Q}= \{\mathcal{Q}_0,\mathcal{Q}_1, \cdots ,\mathcal{Q}_n\} \in \mathbb{R}^{H\times W\times d}$, the decoding process can be conducted as:

$$\hat{\mathcal{Q}} = \mathcal{D}_{\theta_D}(\hat{\mathcal{Z}}, \mathcal{Q}) \tag{7}$$

It is should be emphasized that the encoding and decoding process are all conducted in parallel.

3.5. Multi-Head Convolutional Self-Attention

We call our particular attention "Convolutional Self-Attention"(as shown in Figure 4), which is computed upon feature maps. The convolutional self-attention can be described as mapping a query map and a set of key-value map pairs to an output, where the query map, key maps, value maps, and output are all 3D tensors. Given an input sequential feature maps $\mathcal{U}= \{\mathcal{U}_0,\mathcal{U}_1, \cdots ,\mathcal{U}_n\} \in \mathbb{R}^{H\times W\times d_{model}}$, we apply convolutional sub-networks to generate the query map and paired key-value map of each frame, i.e., $\mathcal{U}'= \{\{\mathcal{Q}_0,\mathcal{K}_0,\mathcal{V}_0\},\{\mathcal{Q}_1,\mathcal{K}_1,\mathcal{V}_1\}, \cdots ,\{\mathcal{Q}_n,\mathcal{K}_n,\mathcal{V}_n\}\} \in \mathbb{R}^{H\times W\times 3}$.

Given a set of $\{\mathcal{Q}_i,\mathcal{K}_i,\mathcal{V}_i\}$ of frame $\mathcal{U}_i$, the attention map $\mathcal{H}_{(i,j)} \in \mathbb{R}^{H\times W\times 1}$ of frame $\mathcal{U}_i$ and $\mathcal{U}_j$ can be generated by applying a compatibility sub-networks $\mathcal{M}_{\theta_M}$ to the query map $\mathcal{Q}_i$ with the corresponding key map $\mathcal{K}_j$, which can be represented as following equation:

$$\mathcal{H}_{(i,j)} = \mathcal{M}_{\theta_M}(\mathcal{Q}_i, \mathcal{K}_j) \tag{8}$$

After getting all the corresponding attention map $\mathcal{H}_{(i)} = \{\mathcal{H}_{(i,1)}, \mathcal{H}_{(i,2)}, \cdots, \mathcal{H}_{(i,n)}\} \in \mathbb{R}^{H\times W\times 1}$, we make a concatenation operation of $\mathcal{H}_{(i)}$ in the third dimensional, and then a SoftMax operation is applied to $\mathcal{H}_{(i)} \in \mathbb{R}^{H\times W\times n}$ along the dimension $d = 3$.

$$\mathcal{H}_{(i)} = SoftMax(\mathcal{H}_{(i)})_d, \quad d = 3 \tag{9}$$

Finally, the output $\hat{\mathcal{V}}_i$ can be obtained with summation of the element wise production with attention map $\mathcal{H}_{(i,j)}$ with the corresponding value map $\mathcal{V}_j$. This operation can be represented as:

$$\mathcal{V}_{(i)} = \sum_{j=1}^{n} \mathcal{H}_{(i,j)}\mathcal{V}_j \tag{10}$$

In order to jointly attend to information from differen representation subspace at different feature space, a multi-head attention is adopted. The process can be view as:

$$\text{MultiHead}(\hat{\mathcal{V}}_i) = \text{Concat}(\hat{\mathcal{V}}_{i_1}, \cdots, \hat{\mathcal{V}}_{i_h}) \tag{11}$$

In this work, we employ $h = 4$ parallel attention layers.

Table 1. Video frame extrapolation: Quantitative evaluation of ConvTransformer with state-of-the-art methods. Red text indicates the best, and blue text indicates the second best performance.

Model	Next frame						Next 3 frames					
	UCF101 [27]		Adobe240fps [28]		Vimeo [35]		UCF101 [27]		Adobe240fps [28]		Vimeo [35]	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DVF [17]	**29.1493**	**0.9181**	**28.7414**	**0.9254**	27.8021	**0.9073**	**26.1174**	**0.8779**	**25.8625**	**0.8598**	24.5277	**0.8432**
MCNet [30]	27.6080	0.8504	28.2096	0.8796	**28.6178**	0.8726	25.0179	0.7766	24.9485	0.7721	**25.4455**	0.7671
Ours	**29.2814**	**0.9205**	**30.4233**	**0.9457**	**30.5161**	**0.9406**	**26.5762**	**0.8801**	**31.5234**	**0.9484**	**25.3802**	**0.8631**

3.6. Synthesis Feed-Forward Network $\mathcal{S}_{\theta_S}$

In order to synthesize the final photo realistic video frames, we construct a frame synthesis feed-forward network, which consists of 2 cascaded sub-networks built upon a U-Net-like structure. The frames states $\mathcal{Q}'$ decoded from previous decoder are fed into SFFN in parallel. This process can be represented as:

$$\hat{\mathcal{X}}_i = \mathcal{S}_{\theta_S}(\mathcal{Q}'_i), \quad i \in [1, n] \tag{12}$$

3.7. Training Loss $\mathcal{L}_{\mathcal{G}_{\theta_G}}$

In this work, we choose the most widely used content loss, i.e., pixel-wise **MSE loss**, to guide the optimizing process of ConvTransformer. The **MSE loss** is calculated as:

The definition of training loss $\mathcal{L}_{\mathcal{G}_{\theta_G}}$ is critical for the performance of ConTransformer. A most widely used pixel-wise **MSE loss**

$$\mathcal{L}_{\mathcal{G}_{\theta_G}} = \frac{1}{N}\sum_{i=1}^{N}\left\|\hat{\mathcal{X}}_i - \mathcal{Y}_i\right\|_2 \tag{13}$$

Here, N is the number of samples, $\hat{\mathcal{X}}_i$ and the $\mathcal{Y}_i$ is the corresponding groundtruth.

4. Experiments and Analysis

In this section, we firstly introduce the training dataset and parameters setting for training process, and then evaluate the proposed algorithm ConvTransformer by comparing with several the-state-of-the-art methods. Finally, several ablation studies are conducted to validate the advantages and effectiveness of each part within ConvTransformer.

4.1. Datasets

To create the trainset of video frame sequence, we leverage the frame sequence from the Vimeo90K datasets [35], which is a newly built high-quality dataset for video frame synthesis. On the other hand, we also exploit several other widely used benchmarks, including UCF101 [27], Sintel [11], REDS [19], HMDB [14] and Adobe240fps [28], for testing.

4.2. Training Details and Parameters Setting

Our ConvTransformer is implemented in Pytorch platform and trained on Nvidia RTX 3090 GPU. In order to guarantee the convergence of ConvTransformer, the Adam optimizer is adopted for training, in which, the initial learning rate is set to 10^{-4} and is reduced with exponential decay, in which, the decay rate is 0.95 while the decaying quantity is 20000. The whole training proceeds for 6×10^5 iteration. Besides, the length of input sequence is 6.

4.3. Comparisons with State-of-the-arts

We compare our finally trained ConvTransformer on several public benchmarks with state-of-the-art video frame interpolation and video frame extrapolation algorithms including DVF [17], MCNet [30], SepConv [21], CyclicGen [16], DAIN [1] and BMBC [22]. For a fair comparison, we implemented and retrained these methods with the same trainset for training ConvTransformer. We evaluate the extrapolation results and interpolation with two widely used image quality metrics: peak signal to noise ratio (PSNR) [10] and structural similarity (SSIM) [31]. Table 1 and Table 2 illustrate the quantitative comparison in terms of PSNR and SSIM of video frame extrapolation and interpolation, respectively. Besides, the visual comparison of synthesized images with zoomed details are represented in Figure 1 and Figure 3.

In Table 1, we represent the quantitative comparisons in terms of extrapolation task. The quantitative result computed on several benchmarks indicates that the proposed ConvTransformer perofrms favorably against previous developed methods. Specifically, compared to the method MCNet, our ConvTransformer achieves 1.9 dB advantage in terms of PSNR on Adobe240fps dataset on Vimeo dataset for Next frame extrapolation. Besides, the results represent that the advantage of ConvTransformer becomes larger in multiple future frames prediction task. For example, ConvTransformer gains a advantage of 5.66 dB in PSNR over the method while it is 1.68 dB in next frame extrapolation on the same benchmark Adobe240fps.

As observed from zoon-in regions in Figure 1, previous methods DVF and MENet produce smoother extrapolation frames, in which, local high-frequency information are suppressed. On the contrary, the proposed ConvTransformer could synthesize more photo realistic frame. Besides, the occlusion maps indicates that ConvTransformer suffers from less occlusion as compared with previous methods.

The quantitative and qualitative results proves that the

Table 2. Video frame interpolation: Quantitative evaluation of ConvTransformer with state-of-the-art methods. Red text indicates the best, and blue text indicates the second best performance.

Model	Sintel [11]		UCF101 [27]		Adobe240fps [28]		HMDB [14]		Vimeo [35]		REDS [19]	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DVF [17]	30.71	0.9303	32.31	0.9454	33.24	0.9613	33.59	0.9469	30.99	0.9379	20.44	0.6460
SepConv-Lf [21]	**31.68**	**0.9470**	32.51	0.9473	**36.36**	**0.9844**	**33.90**	0.9483	33.49	**0.9663**	21.32	0.6965
DAIN [1]	31.37	0.9452	**32.72**	**0.9506**	36.15	**0.9837**	33.89	**0.9487**	**33.95**	**0.9701**	**21.85**	**0.7203**
CyclicGen [16]	**31.54**	0.9104	**33.03**	0.9303	35.60	0.9691	**34.16**	0.9242	33.04	0.9319	21.29	0.5686
CyclicGen-large [16]	31.19	0.9004	32.43	0.9239	34.89	0.9626	33.75	0.9208	32.04	0.9163	20.92	0.5465
BMBC [22]	27.01	0.9223	27.92	0.9412	28.58	0.9569	28.42	0.9384	30.61	0.9629	21.21	**0.7056**
Ours	31.44	**0.9469**	32.48	**0.9504**	**36.42**	**0.9844**	33.37	**0.9492**	**34.03**	0.9637	**21.68**	0.6959

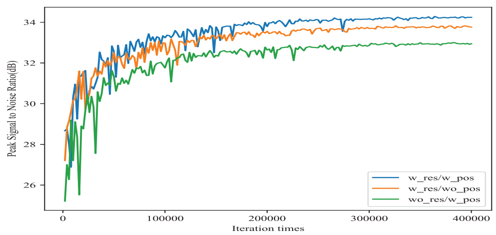

Figure 5. The convergence comparison with ConvTransformer and its degradation variants.

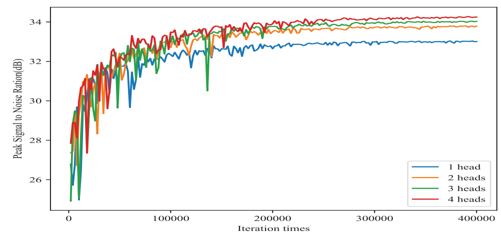

Figure 6. The convergence comparison with ConvTransformer under different head numbers

proposed multi-head convolutional self-attention mechanism can efficiently model the long-range sequential dependence in video frames.

Table 2 and Figure 3 represent the comparison between ConvTransformer and state-of-the-art video frame interpolation methods, i.e. DVF, SepConv, CyclicGen, DAIN and BMBC. The quantitative results in terms of criterion PSNR and SSIM represent that ConvTransformer achieves comparable results against state-of-the-art method DAIN.

The video frame interpolation results prove the better generalization of ConvTransformer, which not only works well on video frame extrapolation, but also suitable for video frame interpolation, as compared with previous recurrent based methods. On the contrary, the state-of-the-art warping based interpolations methods can not be directly used for video frame extrapolation because of the demand of latter frame used for estimating optical flow maps is not provided in video frame extrapolation task.

4.4. Ablation Study

In order to evaluate and justify the efficiency and superiority of each part in the proposed ConvTransformer architecture, several ablation experiments have been conducted in this work. Specifically, we gradually modify the baseline ConvTransformer model and compare their differences.

4.4.1 Investigation of Positional Encoding and Residual Connection

In order to evaluate and justify whether the positional encoding for input sequence is useful, a positional free degradation variant called ConvTransformer-d1 is implemented

Table 3. Comparative results achieved with the ablation of Residaul Connection and Positional Encoding.

Model	Residual Connection	Positional Encoding	PSNR	SSIM
ConvTransformer	✓	✓	**34.2412**	**0.9736**
ConvTransformer-d1	✓	×	**33.7616**	**0.9705**
ConvTransformer-d2	×	✓	32.9461	0.9663

Table 4. Comparative results achieved by ConvTransformer with different head numbers

Model	ConvTransoformer-H-1	ConvTransoformer-H-2	ConvTransoformer-H-3	ConvTransoformer-H-4
PSNR	33.0054	33.7861	**34.0274**	**34.2438**
SSIM	0.9641	0.9695	**0.9714**	**0.9731**

Table 5. Quantitative evaluation of ConvTransformer with variation number of layers in Encoder and Decoder

Model	ConvTransoformer-L-1	ConvTransoformer-L-2	ConvTransoformer-L-3	ConvTransoformer-L-5
PSNR	34.2438	34.4623	**34.4989**	**34.6031**
SSIM	0.9731	0.9741	**0.9744**	**0.9754**

and trained as the same implementation with ConvTransformer. Besides, a degradation variant ConvTrnasformer-d2 removing the residual connection architecture is also modified and gets a full training as ConvTransformer.

The quantitative results depicted in Table 3, convergence cures shown in Figure 5 and vision results shown in Figure 7 indicates that the positional encoding and residual connection is indispensable for ConvTransformer to synthesize naturally high perceptual frames.

4.4.2 Investigation of Multi-Head Numbers Setting

The Multi-Head architecture is adopted for incorporating more information. In order to justify its efficiency of multi-head, a multi-head variation experiments is implemented in this work. The quantitative results represented in Table 4

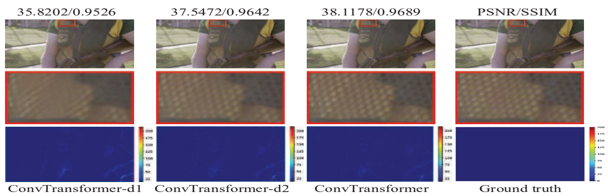

Figure 7. Visual results of ConvTransformer, positional encoding ablated ConvTransformer and ConvTransformer discarding the residual connection.

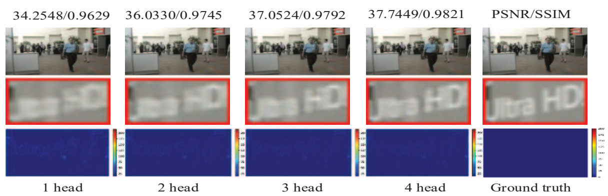

Figure 8. Visual results of ConvTransformer variants with different head numbers.

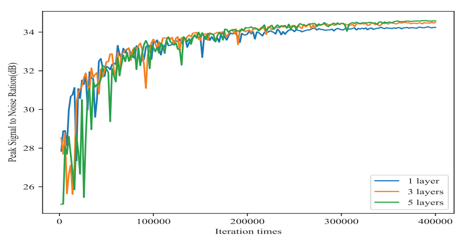

Figure 9. The convergence comparison with ConvTransformer under variant layer numbers.

and qualitative results shown in Figure 8 demonstrate that multi-head architecture is helpful for increasing the representation ability of ConvTransformer.

4.4.3 Investigation of Layer Numbers Setting

The encoder and decoder is built on a sequence layers. It is essential to evaluate the influence of layer numbers setting. Table 5 shows that ConvTransformer performs better with more layers stacked. the core part of ConvTransformer is encoder and decoder, which is build upon stacked layers. It is essential to investigate the influence of layer numbers on the performance of ConvTransformer

4.5. Long-Term Frame Sequence Dependence Analysis

In order to verify whether the proposed multi-head convolutional self-attention could efficiently exploit the long-range sequential dependence within a video frame sequence, we visualize the attention map of decoder query $\mathcal{Q}_i$ on input sequence $\mathcal{X}$ on decoder layer 1. As shown in Figure 10, the correspondent attention map $\mathcal{H}_(i,j)$ normalized in range [0,1] represents the exploiting of input frame $\mathcal{X}_j$ for synthesizing the target frame $\hat{\mathcal{Q}}_i$. It should be emphasized that the pixel value closing to 1 is drew with color blue, while the color yellow means the pixel value close to 0.

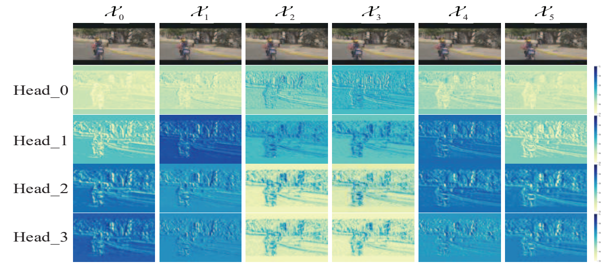

Figure 10. The attention map $\mathcal{H}_i$ of decoding query $\mathcal{Q}_i$ on input sequence $\mathcal{X}$.

As shown in Figure 10, we can find that the Head-1, predominated by yellow, mainly focus the high-frequency details, while Head-3,looked like a blur map, mainly responsible for extracting the low-pass information. Through the vertical comparison with different head, we can draw a conclusion that different head in multi-head convolution self-attention response for modeling different relations, such as background information, foreground information and high-frequent details.

Through the horizontal observing in Head-1, we can find that different position frames contributes differently information for synthesizing target frame. Specifically, frames $\mathcal{X}_2$ and $\mathcal{X}_3$ supply the high-frequent information, while frames $\mathcal{X}_1$ and $\mathcal{X}_4$ provide the background information.

The analysis above and attention maps shown in Figure 10 demonstrate that the proposed multi-head convolutional self-attention can efficiently model the long-term dependence in video frames, and then help the proposed network ConvTransformer synthesize photo-realistic frames that is spatially and temporally related with video frames.

5. Conclusion

In this work, we propose a novel video frame synthesis architecture ConvTransformer, which not only works well on video frame extrapolation, but also interpolates photo-realistic middle frames. We propose a multi-head convolutional self-attention layer to model the long-range spatially and temporally relation of frames in video sequence. The proposed method ConvTransformer is concise, compact and efficient. Extensive quantitative and qualitative evaluations indicate that the proposed solution ConvTransformer performs favorably against existing frame extrapolation and interpolation methods. The successfully implementation of ConvTransformer sheds light for applying it for other video tasks that need to exploit the long-term sequential dependence in video.

References

[1] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3703–3712, 2019. 2, 3, 4, 6, 7

[2] Wenbo Bao, Wei-Sheng Lai, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 2, 3

[3] W. Bao, X. Zhang, L. Chen, L. Ding, and Z. Gao. High-order model and dynamic filtering for frame rate up-conversion. *IEEE Transactions on Image Processing*, 27(8):3813–3826, 2018. 1

[4] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. 3

[5] T. Brooks and J. T. Barron. Learning to synthesize motion blur. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6833–6841, 2019. 1

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020. 3

[7] R. Castagno, P. Haavisto, and G. Ramponi. A method for motion adaptive frame rate up-conversion. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(5):436–446, 1996. 1

[8] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019. 3

[9] J. Flynn, I. Neulander, J. Philbin, and N. Snavely. Deep stereo: Learning to predict new views from the world's imagery. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5515–5524, 2016. 1

[10] Alain Horé and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *International Conference on Pattern Recognition, ICPR 2010, Istanbul, Turkey, 23-26 August 2010*, 2010. 6

[11] Joel Janai, Fatma Guney, Jonas Wulff, Michael J Black, and Andreas Geiger. Slow flow: Exploiting high-speed cameras for accurate and diverse optical flow reference data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3597–3607, 2017. 2, 6, 7

[12] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9000–9008, 2018. 2, 3

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105. Curran Associates, Inc., 2012. 1

[14] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011. 2, 6, 7

[15] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu. High performance visual tracking with siamese region proposal network. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8971–8980, 2018. 1

[16] Yu-Lun Liu, Yi-Tung Liao, Yen-Yu Lin, and Yung-Yu Chuang. Deep video frame interpolation using cyclic frame generation. In *Proceedings of the 33rd Conference on Artificial Intelligence (AAAI)*, 2019. 4, 6, 7

[17] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4463–4471, 2017. 2, 4, 6, 7

[18] Gucan Long, Laurent Kneip, Jose M Alvarez, Hongdong Li, Xiaohu Zhang, and Qifeng Yu. Learning image matching by simply watching video. In *European Conference on Computer Vision*, pages 434–450. Springer, 2016. 2

[19] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 2, 6, 7

[20] S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive convolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2270–2279, 2017. 2

[21] S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive separable convolution. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 261–270, 2017. 2, 4, 6, 7

[22] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation. *arXiv preprint arXiv:2007.12622*, 2020. 2, 3, 4, 6, 7

[23] W. Ren, J. Zhang, X. Xu, L. Ma, X. Cao, G. Meng, and W. Liu. Deep video dehazing with semantic segmentation. *IEEE Transactions on Image Processing*, 28(4):1895–1908, 2019. 1

[24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. 1

[25] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28:802–810, 2015. 2

[26] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2

[27] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *Computer ence*, 2012. 6, 7

[28] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1279–1288, 2017. 2, 6, 7

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3

[30] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. *arXiv preprint arXiv:1706.08033*, 2017. 2, 6

[31] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 2004. 6

[32] Manuel Werlberger, Thomas Pock, Markus Unger, and Horst Bischof. Optical flow guided tv-l 1 video interpolation and restoration. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 273–286. Springer, 2011. 1

[33] Zhanghao Wu, Zhijian Liu, Ji Lin, Yujun Lin, and Song Han. Lite transformer with long-short range attention. *arXiv preprint arXiv:2004.11886*, 2020. 3

[34] H. Xu, Y. Gao, F. Yu, and T. Darrell. End-to-end learning of driving models from large-scale video datasets. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3530–3538, 2017. 1

[35] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019. 2, 6, 7

[36] Zhefei Yu, Houqiang Li, Zhangyang Wang, Zeng Hu, and Chang Wen Chen. Multi-level video frame interpolation: Exploiting the interaction among different levels. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(7):1235–1248, 2013. 1

[37] C. Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *Acm Transactions on Graphics*, 23(3):600–608, 2004. 1