FVIFormer: Flow-Guided Global-Local Aggregation Transformer Network for Video Inpainting

Weiqing Yan *Member, IEEE*, Yiqiu Sun, Guanghui Yue *Member, IEEE*, Wei Zhou, Hantao Liu *Member, IEEE*

***Abstract*—Video inpainting has been extensively used in recent years. Established works usually utilise the similarity between the missing region and its surrounding features to inpaint in the visually damaged content in a multi-stage manner. However, due to the complexity of the video content, it may result in the destruction of structural information of objects within the video. In addition to this, the presence of moving objects in the damaged regions of the video can further increase the difficulty of this work. To address these issues, we propose a flow-guided global-Local aggregation Transformer network for video inpainting. First, we use a pre-trained optical flow complementation network to repair the defective optical flow of video frames. Then, we propose a content inpainting module, which use the complete optical flow as a guide, and propagate the global content across the video frames using efficient temporal and spacial Transformer to inpaint in the corrupted regions of the video. Finally, we propose a structural rectification module to enhance the coherence of content around the missing regions via combining the extracted local and global features. In addition, considering the efficiency of the overall framework, we also optimized the self-attention mechanism to improve the speed of training and testing via depth-wise separable encoding. We validate the effectiveness of our method on the YouTube-VOS and DAVIS video datasets. Extensive experiment results demonstrate the effectiveness of our approach in edge-complementing video content that has undergone stabilisation algorithms.**

I. Introduction

Video inpainting focuses on generating lost or corrupted content in video frames using the available information from the video. The goal is to create a restored video that is not only fully formed and logical, but also seamless. While image inpainting has seen considerable progress as referenced in [12], [20], [26], the challenge in video inpainting lies in the continuous and complex nature of video content, which makes it unsuitable to directly apply techniques developed for image inpainting. Applying image inpainting methods frame by frame can lead to temporal inconsistencies and noticeable flaws in the video. Therefore, for high-quality video inpainting, it is essential to address both the spatial structure of the video frames and their temporal coherence.

In the past few years, deep video inpainting techniques have achieved significant success. Previous video inpainting networks commonly utilized 3D CNNs [35], [6], [38], [13], [28] for the integration of spatiotemporal details. However, these approaches frequently encountered challenges such as constrained receptive fields in both time and space, as well as alignment issues between neighboring frames. Consequently, they exhibited reduced efficacy in capturing distant content. Transformer's exceptional capability in handling long-range interactions has proven it to be well-suited for video inpainting tasks [30], [45], [21]. Both these methodologies, one based on CNN and the other on Transformers, come with their respective strengths and limitations. For instance, the 3D CNN-based approach by Wang et al.[35] excels in reconstructing the general content but lacks in detail refinement and video continuity. In contrast, Ren et al.[30] employed Transformers to better align adjacent features, although the reconstructed objects lacked structural integrity.

On the basis of these two methods, several academics [45], [21], [43], [8] introduce optical flow in the inpainting process. The two methods of Xu et al.[43] and Gao et al. [8] used optical flow guidance to complete video repair according to local feature information in CNN structure. Zhang et al. [45] and Li et al. [21] used Transformer structure under the guidance of optical flow to complete the repair of video content by connecting long-distance information. Due to the varying motion magnitudes between foreground objects and background, the sharing similar motion patterns are more likely to be related. Consequently, the disparities in motion observed in flows can act as a powerful guide, instructing the attention to retrieve content that is more relevant to the task at hand.

In summary, we designed an optical flow restoration module that uses forward and reverse optical flow to complete the estimated residual optical flow information as a guide for subsequent restoration. Guided by the complete optical flow, we use a transformer network to extract relevant and effective features in video frames from both temporal and spatial dimensions to inpaint the missing regions of the target frames. We refer to this process as the Content Inpainting Module. In the temporal Transformer, we use the Efficient Multi-Head Self-Attention (EMSA) [47] on the input frame sequence. Since the input sequence contains non-adjacent video frames, the same target may have a large offset in the sequence. Information interaction within a small receptive field makes it difficult to establish an effective link; therefore, we use a large window to perform inter-frame EMSA to ensure the integrity of the content in the receptive field. The optical

This work was supported by the Youth Innovation Team Project of Higher Education Institutions in Shandong Province No. 2022KJ268, the Science and Technology Innovation Development Plan Projects of Yantai No. 2023JCYJ046, the National Natural Science Foundation of China No. 62001302,62371305.

W. Yan and Y. Sun are with the School of Computer and Control Engineering, Yantai University, Yantai, 261400, China (e-mail: wqyan@tju.edu.cn).

G. Yue is with the School of Biomedical Engineering, Shenzhen University Medical School, Shenzhen University, Shenzhen 518060, China (Corresponding Author: yueguanghui@szu.edu.cn).

H. Liu and W. Zhou are with the School of Computer Science and Informatics, Cardiff University, Cardiff CF24 4AG, United Kingdom (e-mail: liuh35@cardiff.ac.uk; zhouw26@cardiff.ac.uk).

flow information emphasises the contour structure of moving objects within a single video frame. Therefore, we introduce optical flow information before the spatial Transformer to distinguish between moving objects and background regions. Considering that the optical flow information is generated, there may be some errors. Thus, we use a stream weighting operation when introducing the optical flow. It will adaptively control the influence weight of the optical flow information on the recovered content according to the similarity between the optical flow information and the image information. This can reduce the impact of errors in the optical flow information as much as possible. In the spatial Transformer, we utilise video sequences fused with optical flow information on the content of a single frame using an Efficient Multi-Head Self-Attention (EMSA). This process uses the information of the same object or scene to fill in the missing regions of it.

Since Transformer focuses more on the interaction of long-range information, this may make the local structure of the repaired content disparate from the real structure although the texture is reasonable. Therefore, we propose an hourglass-type Structural Rectification Module that use a local binding to interact with local features on the repaired content to restructure the content. Through the processing of this module, the boundary area between the missing and visible regions will be more coordinated, and the transition of the content of the same object there will be more reasonable. In summary, our contributions are:

1) We propose an optical Flow-guided Global-Local Aggregation Transformer Network for video inpainting which exploit the related flow features to guide the corrupted video frame generation via global-local Transformer feature aggregation.
2) We use the idea of optical flow guidance to design Optical-flow Restoration Module as a pre-processing operation to obtain the contour information of moving objects in the video, so that moving objects can be distinguished from the background area. In order to avoid the situation that missing content cannot be found in the target frame, we design a Content Inpainting Module, which uses the information of discontinuous video sequence from the spatial dimension to carry out content repair. Based on the results obtained from the content restoration module, we designed a Structural Rectification Module to adjust the edge information of the missing area by mixing local and global feature extraction, so as to make it more coordinated with the visible content.
3) In the inpainting process, we optimise the sub-attention mechanism via adding the depth-wise separable way to improve the efficiency of the task implementation. We assess the efficacy of our approach using the YouTube-VOS and DAVIS video datasets. The comprehensive experimental results affirm the effectiveness of our method in enhancing the edges of stabilized video content.

II. Related Work

The field of inpainting initially began with images, with the primary objective of filling in missing portions using retrieved or artificially generated content. The concept of video inpainting, which extends this idea to videos, emerged as a natural progression, introducing the additional dimension of time. In video inpainting, the temporal consistency of the video stream becomes crucial, as it allows us to recover information for the corrupted regions not only from the current frame but also from neighboring frames. Conventional video inpainting techniques, exemplified by references like [1], [11], [27], [7], [10], delve into the geometric relationships, such as homography or optical flows, that exist between the damaged areas in the target frames and the intact portions in reference frames. These relationships are harnessed to achieve precise content synthesis and maintain high fidelity. Nonetheless, these methodologies are often plagued by a significant computational burden stemming from the intensive optimization processes involved, which imposes limitations on their practical applicability in real-world scenarios.

Due to the swift progress in deep learning, there has been a proliferation of more streamlined and potent deep learning-driven approaches for video inpainting. These methods have made notable strides in enhancing both the quality and speed of inpainting outcomes. These methods can be broadly categorized into three main categories:optical flow-based methods, 3D convolution methods, and Transformer-based methods.

Approaches utilizing 3D convolutions [3], [16], [24] commonly address missing content by directly incorporating temporal features between adjacent frames through 3D temporal convolution. For instance, the [35] pioneered the development of the initial deep video inpainting network, employing a structure comprising a 3D convolutional network for temporal feature extraction and a 2D convolutional network to restore spatial details. Expanding on this groundwork, the [14] introduced a recurrent 3D-2D network intended to fuse temporal features into the areas within the target frame requiring inpainting. While 3D convolution methods excel in efficiently integrating temporal information, it's noteworthy that 3D CNNs tend to entail relatively higher computational complexities compared to their 2D counterparts. This aspect may impose limitations on the practical application of these techniques.

To address this issue, some researchers have tackled the aggregation of temporal information by framing it as a pixel propagation problem and leveraging optical flow data to depict the evolving relationships between pixels [43], [8], [14], [15], [46], [51]. This spatio-temporal feature aggregation framework, initially applied to video target detection, involves computing the optical flow field between adjacent frames using a pre-trained model, as described by Zhu et al.[50]. They map features from previous frames to the current frame's coordinate space, aggregating them with current frame features through weighted averaging. Wu et al.[39] introduced an Recurrent Neural Network(RNN)-based method for enhancing detection accuracy by aggregating semantic features across frames, taking into account motion trajectories, occlusions, and scene dynamics. Cui et al.[5] proposed the Tf-blender framework, incorporating an attention mechanism for adaptive feature fusion, focusing on the most relevant features for current frame detection. Zhou et al.[48] developed the

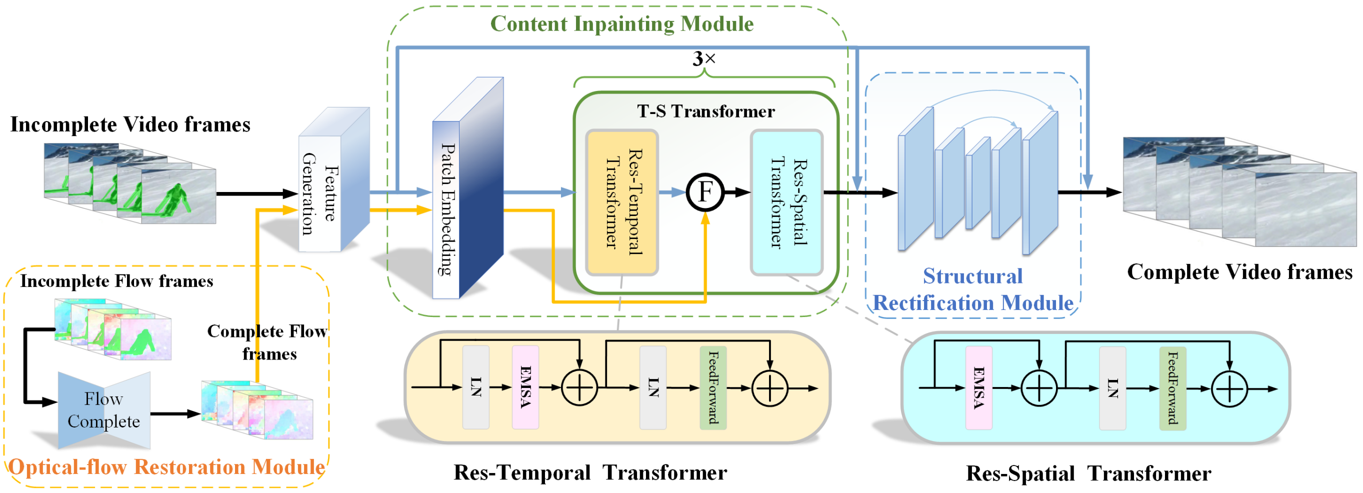

Fig. 1. Overview of the Flow-Guided Global-Local Aggregation Transformer Network for Video Inpainting. It includes 1) Optical-flow Restoration Module(ORM), 2) Content Inpainting Module(CIM), and 3) Structural Rectification Module(SRM). The framework first uses the Optical-flow Restoration Module to repair the missing stream information of the content, then uses the Content Inpainting Module to synthesise the missing regions within the video under the guidance of the optical flow, and finally optimises the details of the seams by means of the Structural Rectification Module.

TransVOD architecture, combining spatial transformers for feature map alignment within frames and temporal transformers to capture motion information across frames. This approach is now widely used in video-related tasks. These approaches typically start by introducing a pre-trained flow completion network to reconstruct the optical flow frames. Subsequently, they employ this restored flow to inpaint the missing regions of neighboring sequence. For instance, in [43], the flow sequence, reconstructed through a coarse-to-fine flow completion network, serves as a guide for incorporating essential pixels into the areas that need to be inpainted. Expanding upon this concept, the [8] addressed the completion of flow edges. The [14] designed flow completion module and an error compensation network guided by error maps, which improves temporal consistency and visual quality in inpainted videos. Furthermore, the [51] tackled spatial misalignments during the propagation of temporal features by leveraging the completed optical flow. While these methods have shown promise, they do have limitations when it comes to aggregating visible content from distant frames due to the inherent nature of optical flow information.

To effectively capture long-distance correspondences, state-of-the-art techniques [18], [19], [23], [25], [40], [44] have adopted Transformer models [33], known for their ability to handle long-term relationships. For instance, Zeng et al. [44] introduced the first transformer-based model for video inpainting. Furthermore, Liu et al. [23] improved the depiction of edge details in the missing region by integrating soft composition operations within the framework of the transformer.

In a related development, Ren et al. [30] introduced an approach named the Discrete Latent Transformer (DLFormer). This approach reformulates the video inpainting task within a discrete latent space, offering a unique perspective on addressing the challenge. While these methods excel at aggregating information from distant parts of the video, they sometimes struggle with maintaining local structural coherence in the inpainted results, which can lead to inconsistencies with reality. Zhou et al.[49] introduced ProPainter, a method that enhances propagation accuracy and efficiency by leveraging previous frame restoration outcomes and current frame data, effectively reducing error accumulation and boosting restoration quality. Wu et al.[41] introduced a semi-supervised technique via the cycle consistency constraints, enabling networks to restore entire videos' corrupted areas with just a single frame's annotated mask.

Although Transformers have demonstrated remarkable performance, their self-attention mechanisms introduce a challenge due to their high computational complexity. When designing a structure for video restoration, it's essential to consider not only the quality of the restoration but also the efficiency of the process. In this context, we introduce a Flow-Guided Global-Local Aggregation Transformer Network that addresses both local and global feature relationships during the restoration process. This approach effectively tackles the computational complexity associated with the self-attention mechanism, offering a more balanced solution for video restoration tasks.

III. Proposed Method

Given a corrupted video sequence with length T $\mathbb{I} = \left\{I_t \in \mathbb{R}^{\mathrm{H}\times\mathrm{W}\times 3} \mid t = 1\dots T\right\}$ and corresponding frame wise binary masks $\mathbb{M} = \left\{M_t \in \mathbb{R}^{\mathrm{H}\times\mathrm{W}\times 3} \mid t = 1\dots T\right\}$, our model output the complete video sequence $\mathbb{Y} = \left\{Y_t \in \mathbb{R}^{\mathrm{H}\times\mathrm{W}\times 3} \mid t = 1\dots T\right\}$. In the following, we will specifically discuss the components of our approach. As shown in Fig. 1, the structure of our network consists of three main modules: Optical-flow Restoration module, Content Inpainting module and Structural Rectification module. For a given masked video sequence I_t, Optical-flow Restoration module inputs its forward and backward optical flows, then, an hourglass network is used to recover the optical flow information in the missing area. Based on completed flows, Content Inpainting

Module propagates the content across video frames, which aggregates high similarity information from neighbouring frames to the target frame. Eventually, Structural Rectification Module interacts the information between the repaired and unmasked content and reconstructs them to a final video sequence $\mathbb{Y}$.

A. *Optical-flow Restoration Module(ORM)*

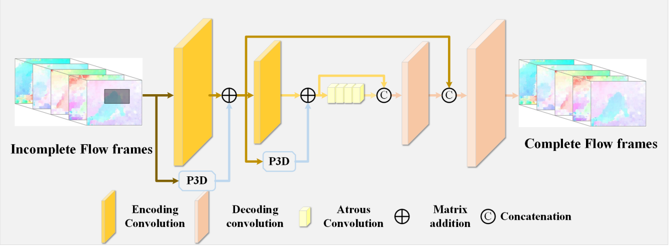

Fig. 2. Structure of Optical-flow Restoration Model. The module uses a two-layer coding and decoding structure and introduces P3D blocks to achieve the aggregation of temporal and spatial flow information during the coding process.

In this section, we will detail the proposed the Optical-flow Restoration Module (ORM). The framework of ORM is presented as Fig. 2. We firstly use the optical flow estimation network Recurrent All-Pairs Field Transforms(RAFT)[31] to estimate the residual optical flow information $\mathbb{F} = \left\{f_t \in \mathbb{R}^{\mathrm{H}\times\mathrm{W}\times 3} \mid t = 1\ldots T\right\}$ of the corrupted video sequence $\mathbb{I}$. Then, we add the Pseudo-3D residual networks (P3D)[29] block to the skip connect to aggregate the features of the flow sequence from both time and space dimensions. Compared with the skip connect, P3D adds one dimensional space and time convolution to enhance the connection of video sequence information. In this module, the residual optical flow information sequence $\mathbb{F}$ is inpainted to obtain the complete optical flow information sequence $\mathbb{F}' = \left\{F_t \in \mathbb{R}^{\mathrm{H}\times\mathrm{W}\times 3} \mid t = 1\ldots T\right\}$. The feature extraction depth of the hourglass repair network is 2, and we take f_t as the input and $\tilde{f}_t^2$ as the output of the encoding of ORM. The encoding process of the optical flow is shown in Eq.1:

$$\begin{aligned} P3D(f_t) &= TC(SC(f_t)) \\ \tilde{f}_t^1 &= P3D(f_t) + Encoder(f_t) \\ \tilde{f}_t^2 &= P3D(\tilde{f}_t^1) + Encoder(\tilde{f}_t^1) \end{aligned} \tag{1}$$

where, $TC()$ is a one-dimensional time convolution, $SC()$ is a two-dimensional space convolution, and they work together to form a P3D block. $Encoder()$ represents the coding layer of the local stream feature aggregation network. In these blocks, the convergent flow characteristics of the target flow are obtained by reducing the time resolution of the flow sequence. As shown in Fig. 2, we decode the local aggregation feature $\tilde{f}_t^{m+1}$ after dilation convolution to complete the stream information inpainting to obtain the complete flow $\mathbb{F}'$. The process of Optical-flow Restoration Model is shown in the following Eq.2.

$$\begin{aligned} \tilde{f}_t^1 &= Decoder(Concat(DC(\tilde{f}_t^2), \tilde{f}_t^2)) \\ F_t &= Decoder(Concat(DC(\tilde{f}_t^1), \tilde{f}_t^1)) \end{aligned} \tag{2}$$

where, $DC()$ represents void convolution, $Concat()$ represents feature cascade operation, $Decoder()$ represents the decoding layer of the local stream feature aggregation network, $\tilde{f}_t^n$ represents the stream feature after n times of encoding, n is the number of coding layers, and t is the t-th frame.

B. *Content Inpainting Module(CIM)*

Once the entire optical flow information has been acquired from the preceding stage, we proceed to inpaint the missing regions in the video based on the optical flow sequence F. Our content inpainting module incorporates both temporal and spatial transformers. He et al.[12] have highlighted the issue of content redundancy arising from processing information solely from local temporal neighbors. To address this concern, we advocate for sequences that encompass information from non-adjacent frames, offering advantages over sequences containing information only from adjacent frames. In this module, we introduce multiple the Res-Spatial Transformer and Res-Temporal Transformer blocks (STTformer) to effectively complete the aggregation of local spatial information and non-local temporal information for content inpainting. In these transformer module, we introduce Efficient Multi-Head Self-Attention (EMSA) [47] to reduce the computational complexity.

Before the content Inpainting module, we downsized the resolution of both the video sequence I_t and the complete optical flow sequence F_t to one-fourth of their original dimensions. Illustrated in Fig. 1, the corrupted I_t and completed F_t is input to the flow-guided feature generation module to propagate the information of I_t from the generated optical flow sequence F_t via convolutional layers, which enhances the accuracy of subsequent operations. Following the feature extraction process, we derived the characteristics of the video sequence as $\hat{I}_t$ and the optical flow feature sequences as $\hat{F}_t$. To facilitate input into the temporal-spatial interleaved Transformer network, we partitioned $\hat{I}_t$ and the completed flow sequence $\hat{F}$ into small patches, and project them into frame token X_t^{token} and F_t^{token} for further processing.

Following that, we present the Content Inpainting Module, including the EMSA, Res-Temporal Transformer, Res-Spatial Transformer.

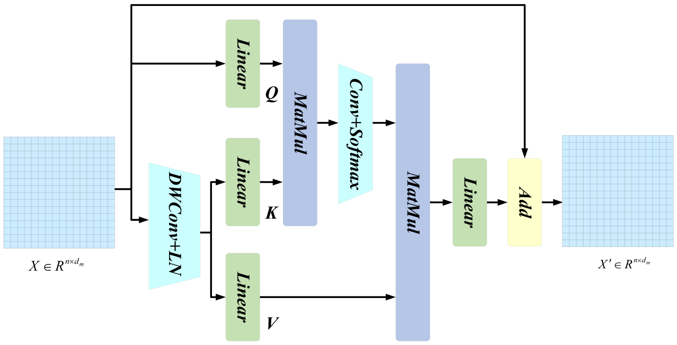

Fig. 3. Efficient Multi-Head Self-Attention. The depth wise separable convolution introduced within the process reduces the dimensionality of the input tokens, which reduces the amount of self-attention operations and improves processing efficiency.

1) Efficient Multi-Head Self-Attention (EMSA): The Multi-Headed Self-Attention (MHSA)[34] exhibits clear advantages compared to conventional sequence modeling techniques, offering capabilities to handle extensive dependencies and facilitate parallel computing. Nonetheless, the substantial consumption of computational and storage resources poses efficiency challenges for MHSA in practical applications. The computational complexity of MHSA is $O(2d_m n^2 + 4d_m^2 n)$, where n denotes the spatial dimension of the input data and d_m denotes the channel size. To address these issues, we introduce the EMSA module (shown in Fig. 3) aimed at mitigating computational and storage demands while maintaining effectiveness.

Step 1: Similar to MHSA, EMSA also takes matrix multiplication to obtain the three values Q, K, V. However, to improve the efficiency of the self-attention operation, a depth wise separable convolution is introduced before using the 2D input token $X_t \in \mathbb{R}^{n \times d_m}$ as a way of splitting the dimension n into the form $h \times w$, (i.e., $\hat{\mathbf{x}} \in \mathbb{R}^{d_m \times h \times w}$)and decreasing the dimensionality of h and w by a factor k. p is a built-in parameter adaptive setting for depth wise separable convolution. The kernel size, step size, and padding are $p+1$, p, and $p/2$, respectively.

Step 2: Reshape a 3D token mapping $\hat{\mathbf{x}} \in \mathbb{R}^{d_m \times h/p \times w/p}$ with reduced width and height into 2D one, i.e.,$\hat{\mathbf{x}} \in \mathbb{R}^{n' \times d_m}$, $n' = h/p \times w/p$. The key K and the value V are then obtained using $\hat{\mathbf{x}}$ also using matrix multiplication. The process is shown in Eq.3:

$$\begin{aligned} Q_t &= \mathrm{Linear}(\mathrm{LN}(X_t)) \\ K_t &= \mathrm{Linear}(\mathrm{LN}(DWConv(X_t))) \\ V_t &= \mathrm{Linear}(\mathrm{LN}(DWConv(X_t))) \end{aligned} \tag{3}$$

Step 3: We connect the output value vectors of the three branches of Q, K, V end to end, and then form the final output through linear transformation.

$$\mathrm{EMSA}(Q,K,V) = \mathrm{IN}(\mathrm{Softmax}(\mathrm{Conv}(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}})))V \tag{4}$$

In this context, $Conv()$ denotes a standard 1×1 convolutional operation. The attention function of each head depends on all keys and queries. However, this configuration diminishes the MHSA's capability to incorporate information from distinct subsets of representations across different positions. To revive the diversity in the attention mechanism, we introduce Instance Normalization[32] (represented as $IN()$), applied to the dot product matrix following the Softmax operation.

The computational complexity of EMSA is $\mathscr{O}\big(\frac{2d_m n^2}{p^2} + 2d_m^2 n(1 + \frac{1}{p^2}) + d_m n \frac{(p+1)^2}{p^2} + \frac{k^2 n^2}{p^2}\big)$, much lower than the computational complexity of the original Multi-headed Self-attention(MHSA), especially at the lower stages where p tends to be higher. As shown in Fig. 3, the output of each EMSA is shown as follows:

$$X_t{}' = X_t + EMSA(LN(X_t)) \tag{5}$$

2) Res-Temporal Transformer: The temporal transformer serves the purpose of facilitating the interaction of distant information within a video, enabling the extraction of information from other video frames when there is no loss area within the target frame. This functionality becomes particularly evident when inpainting moving objects in the video. In the temporal transformer, we execute attentional retrieval in the temporal dimension to gather content from a video sequence. Given that the video sequence includes non-adjacent frames in addition to adjacent frames, a large attention window is employed to prevent potential failures in attentional retrieval caused by significant differences between the content of non-adjacent frames and the target frame. Consequently, we segment the image tokens along the spatial dimension into multiple non-overlapping cubes, each of substantial size with dimensions $H \times W \times T$ (where H represents height, W represents width, and T represents time). The attentional retrieval is performed using EMSA within each cube, facilitating the aggregation of long-range information. The implementation of the temporal transformer is detailed in Equation 6.

$$\begin{aligned} P_K^{token'} &= P_K^{token} + EMSA(LN(P_K^{token})) \\ T_K^{token} &= P_K^{token'} + FFN(LN(P_K^{token'})) \end{aligned} \tag{6}$$

where P_K^{token} consists of the K-frame image features, which are the output of content-fusion module, $LN()$ is the linear normal layer, $EMSA()$ is the efficient self-attention mechanism, and $FFN()$ is the feedforward layer of the Transformer.

3) Res-Spatial Transformer: In the spatial transformer, guided by optical flow, we employ large-size windows to implement the attention mechanism, facilitating information interaction within the video frame. While harnessing the complete optical flow to guide the attention mechanism, a straightforward approach involves directly cascading the optical flow token and the image token. However, this operation presents two challenges. Firstly, errors may arise in the generated optical flow information, potentially leading to misguided judgments regarding the relevant region. Secondly, the internal texture information of the same moving region in the optical flow might vary. Therefore, operating based on the same scale may yield results different from the original content. To address these challenges, as depicted in Fig. 4, we initially use a MLP to assess the similarity between the image token T_K^{token} and the optical flow token W_K^{token} information. The optical flow information W_K^{token} is then weighted based on this similarity, resulting in the weighted optical flow information. Subsequently, T_K^{token} and W_K^{token} are concatenated and input into the spatial transformer. The process of weighted fusion is shown in Eq.7:

$$\begin{aligned} \widehat{W}_K^{token} &= W_K^{token} \times MLP(T_K^{token}, W_K^{token}) \\ R_K &= Concat(\widehat{W}_K^{token}, T_K^{token}) \end{aligned} \tag{7}$$

where MLP() indicates the MLP operation. The enhanced flow $\widehat{W}_K^{token}$ is weighted to enhance the filling capacity of the space Transformer. The spatial transformer is implemented as shown in Eq.8.

$$\begin{aligned} R'_K &= R_K + EMSA(LN(R_K)) \\ S_K &= R'_K + FFN\big(LN(R'_K)\big) \end{aligned} \tag{8}$$

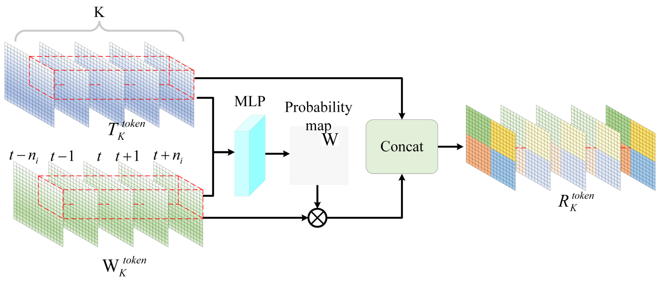

Fig. 4. The process of weighted fusion of optical flow token and image token before entering Spatial transformer.

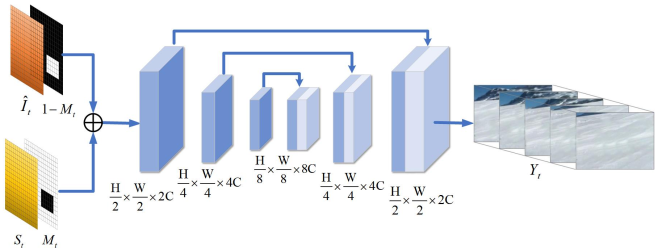
$\hat{I}_t$
$1-M_t$
S_t
M_t
$\frac{H}{2}\times\frac{W}{2}\times 2C$
$\frac{H}{4}\times\frac{W}{4}\times 4C$
$\frac{H}{8}\times\frac{W}{8}\times 8C$
$\frac{H}{4}\times\frac{W}{4}\times 4C$
$\frac{H}{2}\times\frac{W}{2}\times 2C$
Y_t

Fig. 5. Implementation of the Structural Rectification Model. The working process of the structure correction module. Residual links are used in the decoding process to connect the same scale features at the time of encoding.

where, R_K represents the result of cascading the enhanced optical flow feature $\widehat{F}_K^{token}$ with the output T_K^{token} of time Transformer, $Fusion()$ denotes the weighted fusion explained in Eq.7, $LN()$ is the normalization operation, $EMSA()$ is the efficient self-attention mechanism, and $FFN()$ is the feedforward layer of Transformer. S_K^{token} is the output of the space Transformer structure, which is the preliminary repair result of our method.

C. Structural Rectification Module(SRM)

Given that the self-attention mechanism tends to prioritize global information interaction, there is a potential for distortion in the structure of local information. In contrast, CNN are designed to emphasize the interaction of local information. Therefore, as illustrated in Fig. 5, we introduce a 3-layer Encoder-Decoder structure following the transformer block. This addition enhances the coherence of content around the missing regions, refining the outcomes of our inpainting process. The result is a more realistic restoration of partially damaged objects, addressing both structural and textural aspects. The structural correction process is detailed in Equation 9.

$$Y_t = ED(\hat{I}_t \times (1-M_t) + S_t^{token} \times M_t) \tag{9}$$

where $\times$ represents the dot multiplication operation, M_t represents the mask of the damaged area of the video frame, Y_t represents the content of the visible area in the video input, and T_t^{spatial} represents the content generated by the damaged area of the video. $ED()$ represents the working process of the codec. As shown in Fig. 5, the decoding operation takes as input the features of the previous stage and the corresponding coded features in cascade. We verified the effectiveness of the module in the ablation experiment of 4.3.

D. Loss function

We measure the loss using the restored video sequence and its effective feature information.

The first is the image-level loss, which measures the pixel-level difference in the masked area between the inpainted video $\mathbf{Y}_t$ and the true value $\mathbf{I}_t^{gt}$ through the L1 distance:

$$\mathscr{L}_I = \|M_t \times Y_t - M_t \times I_t^{gt}\|_1 \tag{10}$$

The second is feature-level loss, which measures the pixel-level difference between the effective features of the inpainted video sequence and the ground-truth by L1 distance:

$$\mathscr{L}_f = \|FE(Y_t) - \hat{I}_t^{gt}\|_1 \tag{11}$$

where $FE()$ denotes the same operation as feature extraction in Fig. 1, which extracts the valid information in Y_t and aligns the format with $\hat{I}_t^{gt}$ for loss measurement.

In addition to this we introduce, the style loss [9] L_s and and T-PatchGAN loss [4] L_p to supervise the training process. Ultimately our model training loss is designed as a quadruple weighted loss.

$$\mathscr{L} = \lambda_I \mathscr{L}_I + \lambda_f \mathscr{L}_f + \lambda_s \mathscr{L}_s + \lambda_p \mathscr{L}_p \tag{12}$$

where λ_I, λ_f, λ_s and λ_p are the weights of image-level loss, feature-level loss, style loss and T-PatchGAN loss, respectively.

During the training of the model, we set the length of the video frame sequence to 5 and the learning rate to $1e^{-4}$. The Adam optimiser is used for optimisation and the final weights file is obtained after 300 iterations.

IV. Experimental Results

A. Settings

Datasets. We have conducted our evaluation using the Youtube-VOS dataset [42] and the DAVIS dataset [2]. The Youtube-VOS dataset comprises 541 videos, while the DAVIS dataset includes 90 videos, encompassing a wide range of scenes. During the training phase, we utilize the training set from the Youtube-VOS dataset to train our networks effectively. As for masks, during training, we randomly generate irregular masks in proportion to simulate the imperfections of the video content. In the testing process, we use moving rectangular masks and object-like masks respectively for quantitative analysis. For our evaluation metrics, we follow prior research and use the PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index), and LPIPS (Learned Perceptual Image Patch Similarity) [37]. These metrics enable us to comprehensively assess the performance of our method compared to state-of-the-art baselines. VINet[17], STTN[44], FGVC[8], DSTT[22], FFM[23], E2[21] and FGT[45].

B. Comparison of Results

Quantitative results: We have conducted a comprehensive quantitative assessment on both the Youtube-VOS [42] and DAVIS [2] datasets using stationary masks. Our evaluation

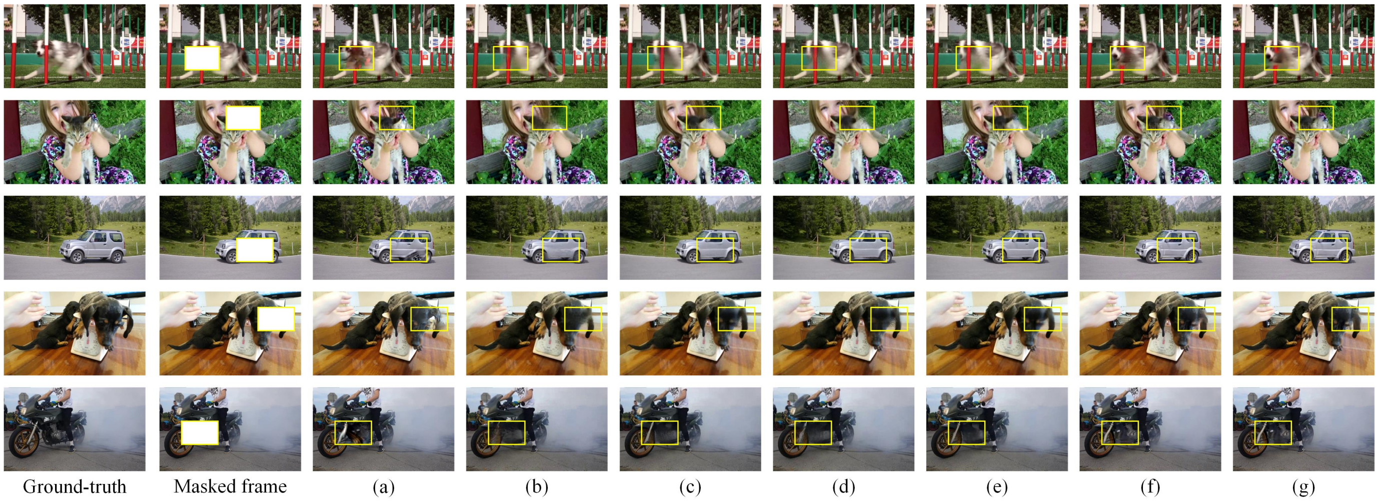

Fig. 6. Comparative results in the DAVIS dataset. (a) The results of FGVC[8], (b) The results of STTN[44], (c) The results of FFM[23], (d) The results of DSTT[22], (e) The results of E2[21], (f) The results of FGT[45], (g) Our results.

TABLE I
QUANTITATIVE RESULTS.

Method	Youtube-VOS			DAVIS-square			DAVIS-object			Efficiency	
	PSNR↑	SSMI↑	LPIPS↓	PSNR↑	SSMI↑	LPIPS↓	PSNR↑	SSMI↑	LPIPS↓	Runtime↓	FPS↑
VINet[17]	30.52	0.9328	0.03449	30.48	0.9128	0.03226	23.28	0.8771	0.01831	0.231	4.329
STTN[44]	31.35	0.9321	0.06582	30.96	0.9136	0.07931	24.05	0.8565	0.01709	0.188	5.588
FGVC[8]	35.17	0.9721	0.02042	35.93	0.9736	0.01635	24.84	0.9091	0.01193	2.162	0.462
DSTT[22]	32.13	0.9343	0.06281	31.63	0.9154	0.07609	24.22	0.8588	0.01674	0.624	1.614
FFM[23]	32.51	0.9369	0.06073	32.07	0.9197	0.07082	24.19	0.8590	0.01679	0.737	1.377
E2[21]	32.79	0.9395	0.05973	32.47	0.9236	0.07275	24.19	0.8598	0.01697	0.189	5.669
FGT[45]	35.33	0.9737	0.01792	35.96	0.9758	0.01514	24.89	0.9183	0.01157	2.611	0.384
Ours	35.33	0.9739	0.01761	35.96	0.9758	0.01512	24.91	0.9193	0.01156	0.497	1.999

involves a comparison with several previous video inpainting methods, including VINet [17], STTN [44], FGVC [8], DSTT [22], FFM [23], E2 [21], and FGT [45].

During the inference process, we have created square mask sets that exhibit continuous motion tracking for both the Youtube-VOS and DAVIS datasets. These masks are designed to evaluate the algorithm's capability to repair damaged video content. The average size of the masks within the square mask set is set to 1/16 of the entire frame. Moreover, to assess each algorithm's ability to remove objects from videos, we have generated mask sets that correspond to the regions of moving objects in the DAVIS dataset. As depicted in Table I, our method outperforms all previous state-of-the-art algorithms across all three quantitative metrics. These outstanding results demonstrate that our approach excels in generating videos with minimal distortion (as indicated by PSNR and SSIM) and more visually plausible content (as indicated by LPIPS).

Qualitative results: Fig. 6 shows the results of repairing the rectangular mask damage video by using different methods. From the example in the third row, we can see that the result generated by the FGVC[8] method is obviously deformed, and the structural shape of the car in the example is destroyed. From the example in the second line, we can see that the results generated by the STTN[44] and DSTT[22] methods will result in the absence of object content, and the part of the cat's ear blocked by the mask will not be filled out. From the example in the fourth line, we can see that the results generated by the E2[21] and FFM[23] methods are fuzzy, and the dog's ear part has obvious distortion. From the example of the first line, we can see that the result generated by the FGT[45] method is ghost-like, and we can see that there are redundant outlines in the area of the dog's head. In summary, our method is able to generate faithful texture and structure information. In addition, we also show the results of our method's object removal operation on video in Fig. 9. It can be seen that the generated results are continuous in time, and there is no obvious ghost and distortion.

Efficiency Comparison: In Table I, we compare the Runtime and Frames Per Second (FPS) metrics for these comparative approaches. The FPS and runtime are derived from the processing of a single video frame. Our analysis is performed on a single 2080Ti GPU, taking into account the complete video restoration process. The experimental data demonstrate that our method achieves comparable runtime efficiency while outperforming the comparison methods across three key evaluation metrics.

Compared with FGT: Our method obtains the better performance in the runtime and FPS under the similar metrics of PSNR, SSMI and LPIPS. In addtion, the propopsed method consider the Structural Rectification Model(SRM) to adjust the edge information of the missing area by integrating local and global feature extraction. As shown in Fig. 7(a) and

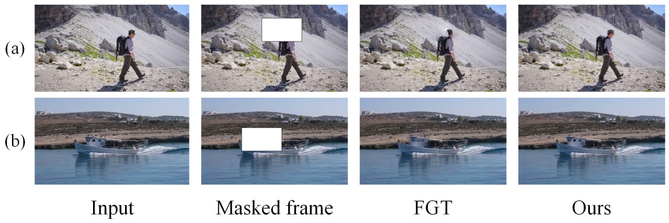

Fig. 7. The visual results of compared with FGT[45].

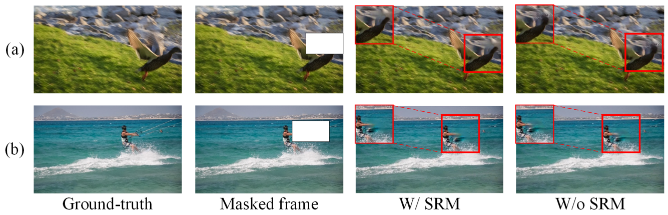

Fig. 8. The result of ablation experiments for the Structural Rectification Module.

(b), when the masked region contains fast moving objects, it can cause more challenges to the contour information of the restoration results. In this case, the results obtained by the FGT method show negative phenomena such as distortion or artefacts, whereas our results distinguish the moving objects from the background region very well, and obtain better visual results in terms of contour integrity and coherence.

C. Ablation Studies

TABLE II
THE RESULTS OF THE ABLATION EXPERIMENT.

Method	DAVIS-object			DAVIS-square		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
W/o SRM	24.78	0.9188	0.01161	35.93	0.9757	0.01520
W/ SRM	24.91	0.9193	0.01156	35.96	0.9758	0.01512

Since E2 et al.[21] demonstrated the effectiveness of intact optical flow for video inpainting, we perform a ablation study on Structural Rectification Model(SRM) to verify its effectiveness. The ablation study is conducted on the DAVIS dataset. As shown in table II, when we remove the structural correction module from the model, the value of the quantitative metrics drops. As can be seen in (a) of Fig. 8, the video recovery deteriorates significantly after removing the module in the comparative methods. The position of the bird's wings in (a) is not well delineated from the scene, and the recovered content at the neck does not match the original content. As can be seen in (b) of Fig. 8, the contours of the objects in the generated region become less obvious after removing this module. The arm region of the person in (b) is covered by a mask and the inpainted contour is not clear after removing this module.

In the Content Inpainting Module, we use the spatio-temporal transformer module with EMSA. To demonstrate the effectiveness, we show ablation experiments for the content repair module. We use a generative network [36] combining ResNet and feature pyramid network to replace the spatio-temporal Transformer structure (STTformer). In addition, the spatio-temporal Transformer structure is adopted but the EMSA is removed. The results are shown in Fig. 10. From this figure, it can be seen that the removal of the STTformer results in a notably incorrect texture for the repaired region. Similarly, when the EMSA is omitted, the contours are inadequately restored, leading to inaccuracies in the texture information repair.

D. Limitation

Video inpainting algorithms face a great challenge when both background regions and dynamic objects appear within a video frame. They have difficulty in discriminating the contents of the two types of regions, which leads to inpainting results in missing content or ghosting. We show two failures of our method in Fig. 11 and list the results of three methods, STTN[44], DSTT[22], and E2[21], which show that the results of the algorithms are difficult to maintain consistency with the original content. We summarise the failure cases and find that the results of video content restoration become unreliable when the following three conditions are simultaneously met: 1) the damaged region of the video contains both the background region and the moving objects; 2) the moving objects are small and move fast; 3) the moving object region has no obvious texture feature information. Failure cases are generally the result of not effectively distinguishing the content of the damaged region from the background, which leads to missing content in the repair results.

V. Conclusion

In this research, we introduce a flow-guided global-local aggregation Transformer network for video inpainting, with a strong focus on enhancing the connections between global and local information within video sequences. Our approach involves the integration of three modules: the Optical-flow Restoration Module, the Content Inpainting Module, and the Structural Rectification Module. These modules work collaboratively to address several limitations observed in earlier methods. We incorporate the concept of optical flow guidance into the design of the Optical-flow Restoration Module, which serves as a preprocessing step aimed at capturing the contour information of moving objects within the video. This allows us to distinguish moving objects from the background area effectively. The Content Inpainting Module, on the other hand, leverages information from discontinuous video sequences in the spatial dimension to perform content restoration. Additionally, we introduce the Structural Rectification Module, which combines local and global feature extraction to better adjust the edge information of the missing area, ensuring its seamless integration with the visible content. Our experimental results clearly demonstrate the outstanding performance of our method, surpassing previous techniques in terms of both quantitative and qualitative evaluations on two widely recognized benchmark datasets.

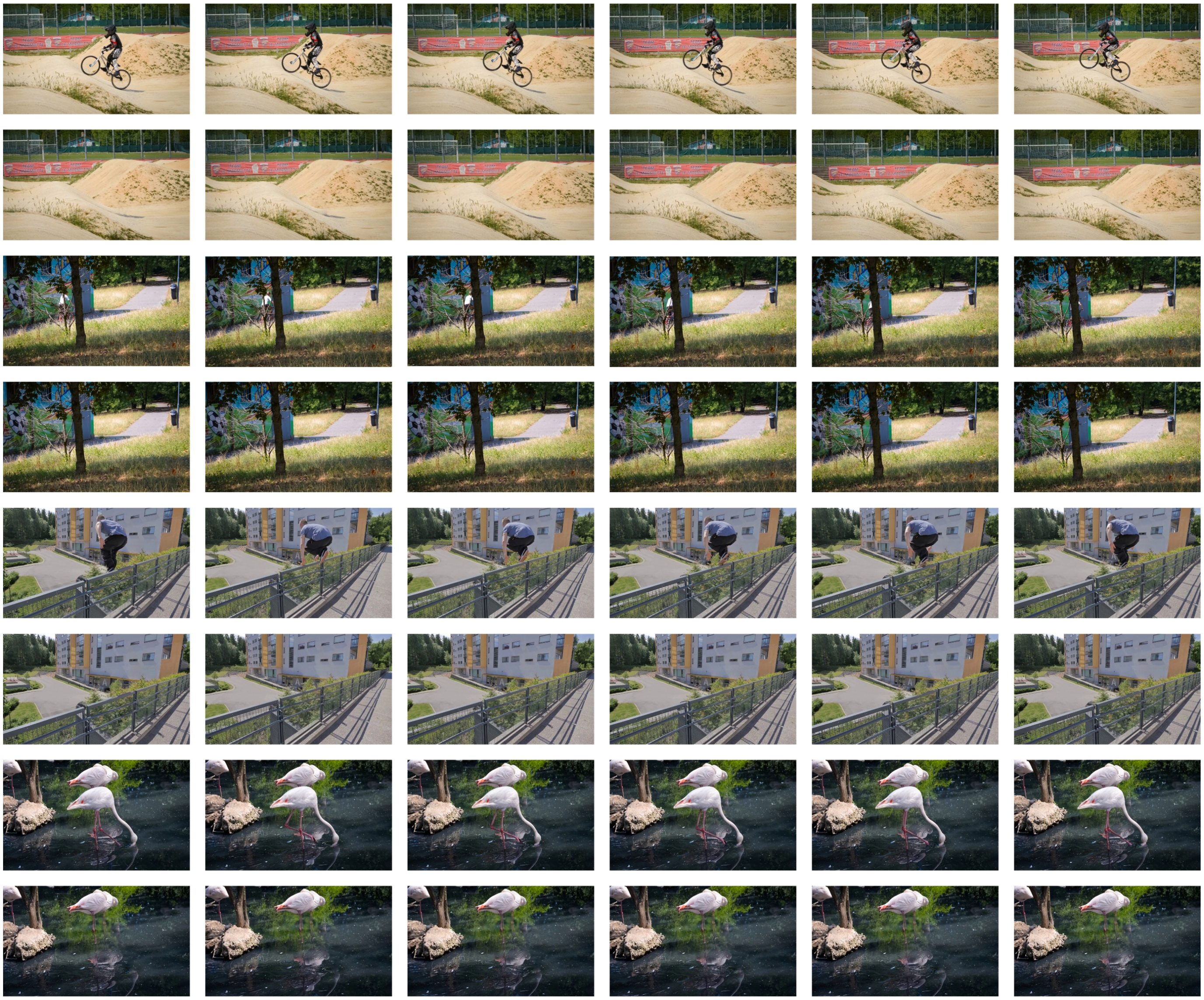

Fig. 9. Object removing via our method. Note: odd-numbered actions input content and even-numbered actions our result.

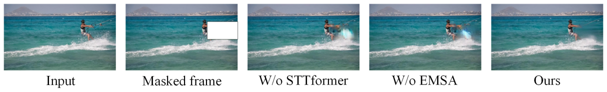

Fig. 10. The result of ablation experiments for the Content Inpainting Module.

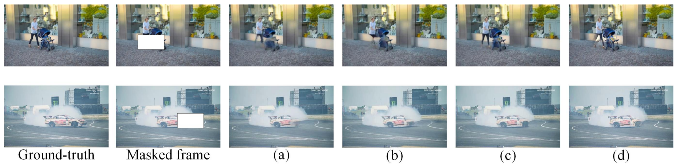

Fig. 11. Failure example. (a) is the result obtained by STTN[44], (b) is the result obtained by DSTT[22], (c) is the result obtained by E2[21], and (d) is our method.

References

[1] Bertalmio, M., Bertozzi, A.L., Sapiro, G.: Navier-stokes, fluid dynamics, and image and video inpainting. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol. 1, pp. I–I. IEEE (2001)

[2] Caelles, S., Montes, A., Maninis, K.K., Chen, Y., Van Gool, L., Perazzi, F., Pont-Tuset, J.: The 2018 davis challenge on video object segmentation. arXiv preprint arXiv:1803.00557 (2018)

[3] Chang, Y.L., Liu, Z.Y., Lee, K.Y., Hsu, W.: Free-form video inpainting with 3d gated convolution and temporal patchgan. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9066–9075 (2019)

[4] Chang, Y.L., Liu, Z.Y., Lee, K.Y., Hsu, W.: Free-form video inpainting with 3d gated convolution and temporal patchgan. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9066–9075 (2019)

[5] Cui, Y., Yan, L., Cao, Z., Liu, D.: Tf-blender: Temporal feature blender for video object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8138–8147 (2021)

[6] Ding, Y., Wang, C., Huang, H., Liu, J., Wang, J., Wang, L.: Frame-recurrent video inpainting by robust optical flow inference. arXiv preprint arXiv:1905.02882 (2019)

[7] Ebdelli, M., Le Meur, O., Guillemot, C.: Video inpainting with short-term windows: application to object removal and error concealment. IEEE Transactions on Image Processing **24**(10), 3034–3047 (2015)

[8] Gao, C., Saraf, A., Huang, J.B., Kopf, J.: Flow-edge guided video completion. In: Computer Vision–ECCV 2020: 16th European Conference,

Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16, pp. 713–729. Springer (2020)

[9] Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576 (2015)

[10] Granados, M., Kim, K.I., Tompkin, J., Kautz, J., Theobalt, C.: Background inpainting for videos with dynamic objects and a free-moving camera. In: Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part I 12, pp. 682–695. Springer (2012)

[11] Granados, M., Tompkin, J., Kim, K., Grau, O., Kautz, J., Theobalt, C.: How not to be seen—object removal from videos of crowded scenes. In: Computer Graphics Forum, vol. 31, pp. 219–228. Wiley Online Library (2012)

[12] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 16000–16009 (2022)

[13] Hu, Y.T., Wang, H., Ballas, N., Grauman, K., Schwing, A.G.: Proposal-based video completion. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16, pp. 38–54. Springer (2020)

[14] Kang, J., Oh, S.W., Kim, S.J.: Error compensation framework for flow-guided video inpainting. In: European Conference on Computer Vision, pp. 375–390. Springer (2022)

[15] Ke, L., Tai, Y.W., Tang, C.K.: Occlusion-aware video object inpainting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14468–14478 (2021)

[16] Kim, D., Woo, S., Lee, J.Y., Kweon, I.S.: Deep blind video decaptioning by temporal aggregation and recurrence. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4263–4272 (2019)

[17] Kim, D., Woo, S., Lee, J.Y., Kweon, I.S.: Deep video inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5792–5801 (2019)

[18] Lee, S., Oh, S.W., Won, D., Kim, S.J.: Copy-and-paste networks for deep video inpainting. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 4413–4421 (2019)

[19] Li, A., Zhao, S., Ma, X., Gong, M., Qi, J., Zhang, R., Tao, D., Kotagiri, R.: Short-term and long-term context aggregation network for video inpainting. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16, pp. 728–743. Springer (2020)

[20] Li, W., Lin, Z., Zhou, K., Qi, L., Wang, Y., Jia, J.: Mat: Mask-aware transformer for large hole image inpainting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10758–10768 (2022)

[21] Li, Z., Lu, C.Z., Qin, J., Guo, C.L., Cheng, M.M.: Towards an end-to-end framework for flow-guided video inpainting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 17562–17571 (2022)

[22] Liu, R., Deng, H., Huang, Y., Shi, X., Lu, L., Sun, W., Wang, X., Dai, J., Li, H.: Decoupled spatial-temporal transformer for video inpainting. arXiv preprint arXiv:2104.06637 (2021)

[23] Liu, R., Deng, H., Huang, Y., Shi, X., Lu, L., Sun, W., Wang, X., Dai, J., Li, H.: Fuseformer: Fusing fine-grained information in transformers for video inpainting. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 14040–14049 (2021)

[24] Liu, R., Li, B., Zhu, Y.: Temporal group fusion network for deep video inpainting. IEEE Transactions on Circuits and Systems for Video Technology **32**(6), 3539–3551 (2021)

[25] Liu, R., Weng, Z., Zhu, Y., Li, B.: Temporal adaptive alignment network for deep video inpainting. In: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, pp. 927–933 (2021)

[26] Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L.: Repaint: Inpainting using denoising diffusion probabilistic models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11461–11471 (2022)

[27] Matsushita, Y., Ofek, E., Ge, W., Tang, X., Shum, H.Y.: Full-frame video stabilization with motion inpainting. IEEE Transactions on pattern analysis and Machine Intelligence **28**(7), 1150–1163 (2006)

[28] Ouyang, H., Wang, T., Chen, Q.: Internal video inpainting by implicit long-range propagation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 14579–14588 (2021)

[29] Qiu, Z., Yao, T., Mei, T.: Learning spatio-temporal representation with pseudo-3d residual networks. In: proceedings of the IEEE International Conference on Computer Vision, pp. 5533–5541 (2017)

[30] Ren, J., Zheng, Q., Zhao, Y., Xu, X., Li, C.: Dlformer: Discrete latent transformer for video inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3511–3520 (2022)

[31] Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16, pp. 402–419. Springer (2020)

[32] Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)

[33] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

[34] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

[35] Wang, C., Huang, H., Han, X., Wang, J.: Video inpainting by jointly learning temporal structure and spatial details. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 5232–5239 (2019)

[36] Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8798–8807 (2018)

[37] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)

[38] Woo, S., Kim, D., Park, K., Lee, J.Y., Kweon, I.S.: Align-and-attend network for globally and locally coherent video inpainting. arXiv preprint arXiv:1905.13066 (2019)

[39] Wu, H., Chen, Y., Wang, N., Zhang, Z.: Sequence level semantics aggregation for video object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9217–9225 (2019)

[40] Wu, Z., Sun, C., Xuan, H., Zhang, K., Yan, Y.: Divide-and-conquer completion network for video inpainting. IEEE Transactions on Circuits and Systems for Video Technology (2022)

[41] Wu, Z., Xuan, H., Sun, C., Guan, W., Zhang, K., Yan, Y.: Semi-supervised video inpainting with cycle consistency constraints. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 22586–22595 (2023)

[42] Xu, N., Yang, L., Fan, Y., Yue, D., Liang, Y., Yang, J., Huang, T.: Youtube-vos: A large-scale video object segmentation benchmark. arXiv preprint arXiv:1809.03327 (2018)

[43] Xu, R., Li, X., Zhou, B., Loy, C.C.: Deep flow-guided video inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3723–3732 (2019)

[44] Zeng, Y., Fu, J., Chao, H.: Learning joint spatial-temporal transformations for video inpainting. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16, pp. 528–543. Springer (2020)

[45] Zhang, K., Fu, J., Liu, D.: Flow-guided transformer for video inpainting. In: European Conference on Computer Vision, pp. 74–90. Springer (2022)

[46] Zhang, K., Fu, J., Liu, D.: Inertia-guided flow completion and style fusion for video inpainting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 5982–5991 (2022)

[47] Zhang, Q., Yang, Y.B.: Rest: An efficient transformer for visual recognition. Advances in neural information processing systems **34**, 15475–15485 (2021)

[48] Zhou, Q., Li, X., He, L., Yang, Y., Cheng, G., Tong, Y., Ma, L., Tao, D.: Transvod: end-to-end video object detection with spatial-temporal transformers. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)

[49] Zhou, S., Li, C., Chan, K.C., Loy, C.C.: Propainter: Improving propagation and transformer for video inpainting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10477–10486 (2023)

[50] Zhu, X., Wang, Y., Dai, J., Yuan, L., Wei, Y.: Flow-guided feature aggregation for video object detection. In: Proceedings of the IEEE international conference on computer vision, pp. 408–417 (2017)

[51] Zou, X., Yang, L., Liu, D., Lee, Y.J.: Progressive temporal feature alignment network for video inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16448–16457 (2021)