Depth-guided Deep Video Inpainting

Shibo Li, Shuyuan Zhu, *Member, IEEE*, Yao Ge, Bing Zeng, *Fellow, IEEE*,
Muhammad Ali Imran, *Fellow, IEEE*, Qammer H. Abbasi, *Senior Member, IEEE*, and Jonathan Cooper

***Abstract*—Video inpainting aims to fill in missing regions of a video after any undesired contents are removed from it. This technique can be applied to repair the broken video or edit the video content. In this paper, we propose a depth-guided deep video inpainting network (DGDVI) and demonstrate its effectiveness in processing challenging broken areas crossing multiple depth layers. To achieve our goal, we divide the inpainting into depth completion, content reconstruction, and content enhancement. Three corresponding modules are designed to implement a process-flow. Firstly, we develop a depth completion module based upon the spatio-temporal Transformer which is used to obtain the completed depth information for each video frame. Secondly, we design a content reconstruction module to generate initially inpainted video. With this module, the contents of the missing regions are composed via the depth-guided feature propagation. Thirdly, we construct a content enhancement module to enhance the temporal coherence and texture quality for the inpainted video. All of proposed modules are jointly optimized to guarantee the high inpainting efficiency. The experimental results demonstrate that our proposed method provides better inpainting results, both qualitatively and quantitatively, compared with the previous state-of-the-art. The code is available at https://github.com/lishibo888/DGDVI.**

***Index Terms*—Video inpainting, depth completion, depth-guided content reconstruction, content enhancement.**

I. Introduction

VIDEO inpainting is a popular restoration technique that is used to complete the damaged video or edit the video with removed contents. It was initially implemented based upon image inpainting and further developed by introducing the temporal information of videos to achieve high inpainting performance, generating visually consistent and coherent video contents.

Although numerous of video inpainting methods [1]–[5] have been proposed in the past, there are still challenges in the design of an effective scheme. Notably, effectively composing the missing regions crossing different depth layers, such as foreground and background layers of the content, remains a significant challenge in video inpainting. Due to the lack of cues to identify layers, foreground and background reference information aliasing frequently happens when information is propagated from reference region to target region for content construction, resulting in blurred edges and details. This aliasing also induces spatial and temporal incoherence between composed frames, generating low-quality inpainted videos.

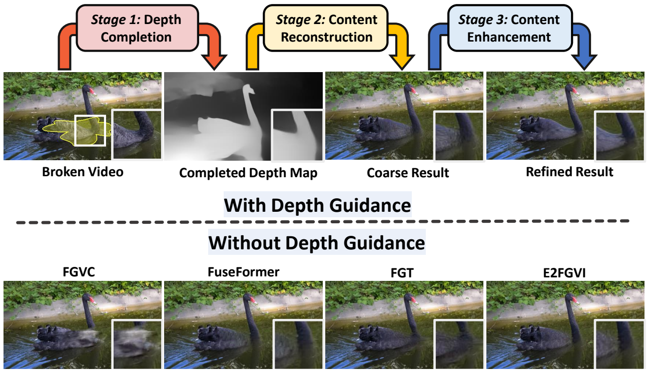

Fig. 1. Pipeline of our proposed depth-guided video inpainting approach. Our approach use the predicted depth map to guide the content reconstruction for missing regions. Compared with the approaches without depth guidance, our method can generate more reliable contents for missing regions that cross foreground and background.

Over the past years, some methods [1], [5], [6] have been proposed to generate spatial and temporal coherent results by introducing optical flow as guidance. These methods estimate optical flow for missing regions and propagate reference information from frame to frame guided by the flow to construct contents for missing regions. The flow-based methods can be implemented in either pixel domain or feature domain, via content propagation or feature propagation throughout the video for completion. Accurate optical flow is crucial for information propagation in these methods. However, the flow often changes dramatically over a long duration, which makes accurate flow estimation for all missing regions over the whole video difficult when the long-range cues are needed. Meanwhile, estimation error often happens and is propagated during flow estimation, especially on the regions crossing foreground and background layers. The existence of this error will result in information propagation error, thus limiting inpainting efficiency.

To solve the above problem, we adopt depth rather than optical flow to guide the information propagation for video inpainting. Compared with flow, depth is temporal invariant over the whole video, which makes it much easier to be predicted for missing regions. In addition, using depth information can effectively distinguish the foreground and background for the video frame. This indicates that adopting it to guide information propagation may potentially solve the reference information aliasing problem.

In a previous study [7], depth is used to implement the warping of reference region to the target broken region, where the reference region is offered by an external image sharing scene contents with the target image. The warped reference region then offers the scene-consistent information for the

S. Li, S. Zhu, and B. Zeng are with School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China.

S. Li is also with James Watt School of Engineering, University of Glasgow, Glasgow, UK.

Y. Ge, M. Imran, Q. Abbasi and J. Cooper are with James Watt School of Engineering, University of Glasgow, Glasgow, UK.

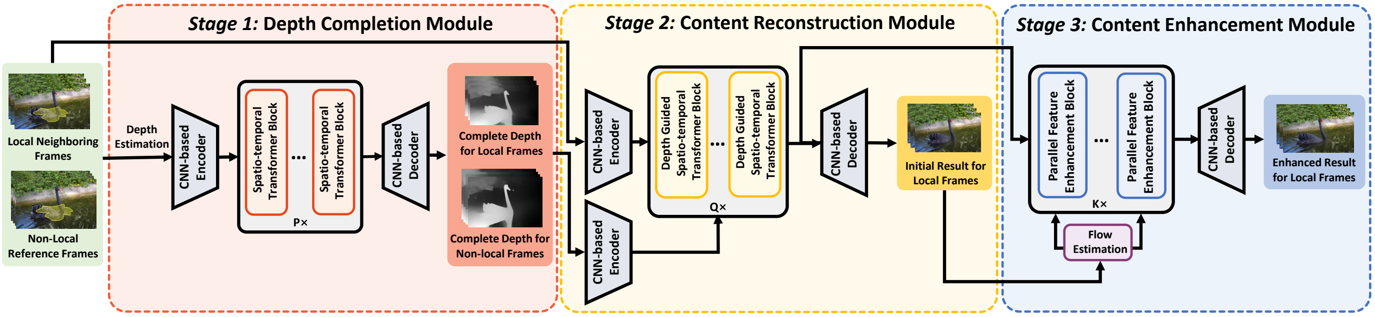
Stage 1: Depth Completion Module
Stage 2: Content Reconstruction Module
Stage 3: Content Enhancement Module
Local Neighboring Frames
Non-Local Reference Frames
Depth Estimation
CNN-based Encoder
Spatio-temporal Transformer Block
...
Spatio-temporal Transformer Block
P×
CNN-based Decoder
Complete Depth for Local Frames
Complete Depth for Non-local Frames
CNN-based Encoder
CNN-based Encoder
Depth Guided Spatio-temporal Transformer Block
...
Depth Guided Spatio-temporal Transformer Block
Q×
CNN-based Decoder
Initial Result for Local Frames
Parallel Feature Enhancement Block
...
Parallel Feature Enhancement Block
K×
Flow Estimation
CNN-based Decoder
Enhanced Result for Local Frames

Fig. 2. Framework of the proposed depth-guided deep video inpainting (DGDVI).

completion of the broken region. Additionally, in [8], the depth information acquired from Lidar is used to guide the fusion of multiple source videos, to generate an inpainted clear video without undesired traffic agents. In contrast to previous work, we aim to use depth to guide the information prorogation for the construction of contents for the broken video. We do not use it to either align video frames or fuse videos to implement inpainting.

To achieve effective video inpainting, we propose a depth-guided method in this work and implement it in three stages, including the depth completion, content reconstruction, and content enhancement, as illustrated in Fig. 1. We design three corresponding modules for these stages, to predict the depth of video, compose content for the missing region, and enhance the composed content, respectively. These modules are subsequently used to construct our depth-guided deep video inpainting network (DGDVI), as illustrated Fig. 2. Our proposed method aims to achieve high robustness and performance for the challenging inpainting scenes, especially for the filling of region crossing different depth layers. Our contributions are summarized as follows:

- We propose a video inpainting method with the guidance of depth. The depth information is adopted to guide the information propagation over the video, composing reasonable and reliable results, especially for the completion of multi-layered region.
- We construct a depth completion module to predict the completed depth for the broken video by using both local and non-local spatio-temporal reference information.
- We design a content reconstruction module to generate contents for missing regions with the guidance of depth, solving the content aliasing problem.
- We develop a content enhancement module with our proposed parallel feature enhancement network to enhance the temporal coherence and texture quality for the video, guaranteeing to achieve high inpainting quality.

II. Related Work

A. Deep Learning-based Image Inpainting

The deep learning-based image inpainting has demonstrated impressive performance over the past few years. The convolution neural networks (CNN) based methods were firstly applied to image inpainting. For instance, Pathak *et al.* [9] introduced the generative-adversarial network [10] to image inpainting and achieved good results. In addition, the advanced modules or learning strategies were proposed to produce high-quality inpainted images, including the contextual attention [11]–[14], partial convolution [15], gated convolution [16] and Fourier convolution [17]. In these methods, Yu *et al.* [11] introduced a contextual attention module to implement the coarse-to-fine generative image inpainting. Liu *et al.* [15] designed an image inpainting network using the partial convolutions. Yu *et al.* [16] constructed an inpainting network based on the gated convolution to selectively integrate valid information collected from the surrounding regions to compose content for the missing region. Recently, Suvorov *et al.* [17] built an image inpainting network using the Fourier convolution to obtain wide receptive field so that the model can implement the large mask inpainting.

In order to generate more reliable contents, some methods [18]–[24] introduced intermediate clues to guide the content reconstruction for missing regions. More specifically, Nazeri *et al.* [18] constructed the Edgeconnect network to predict the edges for the missed contents. The predicted edges are used to guide the filling of missing regions with a completion network. Xiong *et al.* [19] developed a foreground contour completion network to predict foreground contour for the missing regions and built a CNN-based image completion network to generate contents with the guidance of the predicted foreground contour. Ren *et al.* [20] proposed a structure reconstruction CNN model to complete the missing structure information of image and also designed a texture generator to yield image details according to the reconstructed structures. Wu *et al.* [21] constructed a two-staged generative model for image inpainting, which firstly accurately predicts the structural information of the missing region based on local binary pattern learning and subsequently builds a structure-guided image inpainting network using spatial attention. Song *et al.* [22] developed a segmentation prediction model to obtain the segmentation information for the missing regions and then built a segmentation-guided image inpainting network to generate the semantic consistent contents. Moreover, instead of one-way semantic guidance, Zhang *et al.* [23] established a semantic-guided image inpainting framework in which the semantic segmentation guides image inpainting and also receives feedback from image inpainting to generate more reliable inpainting results. Additionally, Sun *et al.* [24] proposed a deep network which learns to decompose a complex mask area into several basic mask types and inpaints the damaged image in a patch-wise manner to enhance the inpainting robustness.

Besides the CNN-based methods, the Transformer-based image inpainting approaches [25]–[27] also achieve remarkable performance. For instance, Yu *et al.* [25] proposed a bidirectional autoregressive Transformer for image inpainting. Li *et al.* [26] proposed a mask-aware Transformer to repair the image with large missing area. Dong *et al.* [27] designed a Transformer model to restore the low-resolution structure for the broken image and also built a Fourier CNN model to generate textures for the missing regions with the guidance of the up-sampled structure.

B. Deep Learning-based Video Inpainting

In recent years, the application of deep learning, especially the CNN-based methods, to video inpainting has also demonstrated good performance. For instance, Lee *et al.* [28] constructed a deep frame alignment network to aggregate cues collected from reference frames to inpaint target frame. To obtain both spatial and temporal cues for inpainting, some methods [29]–[32] employed 3D convolution to construct the deep video inpainting network. Specifically, Wang *et al.* [29] and Kim *et al.* [30] designed the CNN models for inpainting by using both 3D and 2D convolutions to collect spatial and temporal information. Chang *et al.* [31] built a temporal PatchGAN model based on the proposed 3D gated convolution for free-form video inpainting.

Due to the limited spatio-temporal receptive field of the 3D convolution, using 3D convolution often results in inconsistent visual artifacts in the inpainted videos. To solve this problem, some approaches [1], [6], [33]–[35] adopted optical flow as the guidance to propagate cues from the neighboring frames to target frame for content construction. For example, Xu *et al.* [6] firstly proposed a deep flow completion network (DFC-Net) to estimate the flow for the missing region, facilitating the propagation and composition of content in pixel domain. Zou *et al.* [33] used DFC-Net to predict the flow and employed the flow-guided convolution to propagate the reference cues for content construction in feature domain. To obtain the reliable flow for the generation of temporally coherent results, Gao *et al.* [1] employed Edgeconnect [18] to predict the edges of missing contents and then used the edge information to guide the completion of flow. Then, based on the motion information of object, Zhang *et al.* [34] introduced inertia prior to estimate optical flow so as to guarantee using the generated flow can produce good inpainting results. Recently, Kang *et al.* [35] proposed an error compensation method to improve the prediction accuracy of flow for the implementation of high-efficiency flow-guided inpainting.

In addition to the CNN-based method, the Transformer-based video inpainting was also proposed. Based on vision Transformer [36], Zeng *et al.* [2] firstly developed a spatial-temporal Transformer model (STTN) for video inpainting. To improve STTN, Liu *et al.* [3] built FuseFormer model to generate fine-grained contents by using overlapped patch embeddings. Additionally, Liu *et al.* [37] constructed a spatial-temporal attention scheme, implementing the spatial propagation and temporal propagation with two different attention blocks to compose contents. Masum *et al.* [38] constructed an end-to-end network based on axial attention-based style Transformer to achieve consistent video inpainting. To introduce the flow-based guidance into the Transformer-based model, Li *et al.* [4] adopted flow-guided convolution for short-term propagation across neighboring frames and used Transformer model to implement the long-term spatio-temporal propagation for the generation of inpainted videos. Additionally, Zhang *et al.* [5] designed a flow-guided Transformer model to fuse cues to produce high-quality results.

III. Proposed Method

A. Overview

Our proposed DGDVI method is implemented in three stages, including depth completion, content reconstruction and content enhancement, to complete broken videos, especially the one with multi-layered contents. Three corresponding modules are designed for these stages and are jointly optimized for the implementation of our proposed model.

In our work, given a broken video that contains N frames $\{X_1, X_2, \ldots, X_N\}$, where $X_i \in \mathbb{R}^{H\times W\times 3}$, we firstly divide the frames into several local frame groups. Each local frame group, denoted as $\mathbf{X}_l$, composed by N_l frames that are obtained by performing a temporal window on the video to select frames. Meanwhile, we construct one non-local frame group, denoted as $\mathbf{X}_{nl}$, that consists of N_{nl} frames obtained by uniformly sampling the video frames with a given step-size. With $\mathbf{X}_l$ and $\mathbf{X}_{nl}$, we compose $\mathbf{X}_{in} = \{\mathbf{X}_l, \mathbf{X}_{nl}\}$.

Then, in the depth completion and content reconstruction stages, we use $\mathbf{X}_{in}$ to produce a rough inpainting result for $\mathbf{X}_l$. Note that $\mathbf{X}_{in}$ consists of local and non-local frames. The employment of non-local frames to produce the inpainted local frames aims at introducing long-range spatio-temporal context to achieve high quality. Finally, we just use the frames of $\mathbf{X}_l$ to enhance the local temporal coherence for it and obtain a refined result in the content enhancement stage.

B. Depth Completion

The structure, shape and contour cues can be clearly indicated in depth, which makes it be potentially used to enhance the quality of reconstructed contents. In this work, we design the depth completion module to predict depth for damaged video. The predicted depth is then used to guide the content reconstruction. This module is constructed based on the spatio-temporal Transformer with multi-head self-attention and can be used to obtain the local and non-local depth dependencies to generate completed depth.

Given $\mathbf{X}_{in} = \{\mathbf{X}_l, \mathbf{X}_{nl}\} \in \mathbb{R}^{(N_l+N_{nl})\times H\times W\times 3}$, we use a pre-trained depth estimation network [39] to obtain depth information for the available image regions but leave the other regions of depth empty. Assuming that $\mathbf{D}_{in} \in \mathbb{R}^{(N_l+N_{nl})\times H\times W}$ is composed of the incomplete depth of $\mathbf{X}_{in}$, the proposed depth completion module is designed to generate complete depth for $\mathbf{X}_{in}$ based on $\mathbf{D}_{in}$.

Inspired by [2], [3], we construct the depth completion module based on the CNN-Transformer hybrid architecture that enables the module to generate accurate and temporally consistent depth information. In addition, the depth completion is implemented in three steps, including feature extraction, feature propagation and depth construction.

Specifically, the features of $\mathbf{D}_{in}$ are firstly extracted by a CNN-based context encoder and converted into token embeddings. Then, the token embeddings are fed into a stack of spatio-temporal Transformer blocks to complete the feature propagation via token updating. Finally, the updated tokens are converted back into features by a token-to-patch module and a CNN decoder is applied to produce the predicted depth information $\mathbf{D}_c \in \mathbb{R}^{(N_l+N_{nl})\times H\times W}$.

1) Feature Extraction: We firstly concatenate $\mathbf{D}_{in}$ with the corresponding inpainting mask and then feed them into a CNN-based encoder which consists of five convolutional layers to obtain the features $\mathbf{E}_{dep}$ where the channel number is C_{dep} and the size of feature map is $H/4 \times W/4$. Then, the features $\mathbf{E}_{dep}$ are converted into tokens so that they can be fed into the consequent Transformer blocks. In this work, the soft split operation (SS) [3] is applied to split $\mathbf{E}_{dep}$ into overlapped patch embeddings to form the tokens $\mathbf{Z}^0_{dep}$ as

$$\mathbf{Z}^0_{dep} = \mathrm{SS}(\mathbf{E}_{dep}). \tag{1}$$

With the soft split operation, the temporally-correlated features extracted from depth are converted into overlapped token embeddings so that we can build up the spatio-temporal correlation among tokens for the updating of them during feature propagation.

2) Feature Propagation: We adopt feature propagation [2]–[4] to propagate the features of depth from available regions to missing regions in the incomplete depth so that we can complete depth. The spatio-temporal Transformer blocks are employed to facilitate this propagation according to the short-long term dependencies and contextual cues in both feature and temporal domains.

To guide the feature propagation, the multi-head self-attention (MSA) [36], [40] is adopted in the spatio-temporal Transformer block (STTB) [2], [3] to construct the spatio-temporal dependencies within tokens. To implement the MSA mechanism, the input tokens are firstly transformed into the query, key and value vectors, and then are split into multiple heads to extract more diverse and expressive representations than only using single head. For each head, its attention map is obtained by calculating the attention scores between its query and key vectors, capturing multiple relationships between multiple tokens. With the obtained attention map, we can assign appropriate attention weights to relevant values so as to update the tokens in feature and temporal domains.

Assuming that there are k heads in MSA, the updated value $\hat{\mathbf{V}}_k$ for each head is calculated as

$$\hat{\mathbf{V}}_k = \mathrm{MSA}(\mathbf{Q}_k, \mathbf{K}_k, \mathbf{V}_k) = \frac{\mathit{softmax}(\mathbf{Q}_k\mathbf{K}_k^T)}{\sqrt{d_k}}\mathbf{V}_k, \tag{2}$$

where $\mathbf{Q}_k$, $\mathbf{K}_k$, and $\mathbf{V}_k$ are the query, key and value vectors for each head, respectively, and d_k is the feature dimension of $\mathbf{Q}_k$ and adopted as a scaling factor. With the application of MSA, the module can update the tokens for broken regions during feature propagation, guaranteeing to generate accurate depth information.

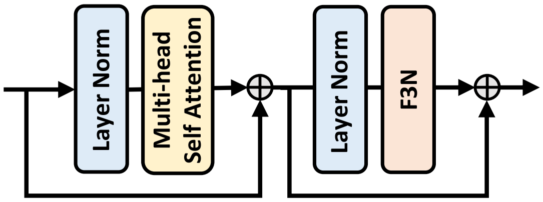

Fig. 3. Architecture of spatio-temporal Transformer block (STTB).

Based on MSA, we construct STTB as illustrated in Fig. 3 and it is implemented as

$$\begin{aligned} \mathbf{Z}'^n_{dep} &= \mathrm{MSA}(\mathrm{LN}(\mathbf{Z}^{n-1}_{dep})) + \mathbf{Z}^{n-1}_{dep} \\ \mathbf{Z}^n_{dep} &= \mathrm{F3N}(\mathrm{LN}(\mathbf{Z}'^n_{dep})) + \mathbf{Z}'^n_{dep}, \end{aligned} \tag{3}$$

where $\mathbf{Z}^{n-1}_{dep}$ denote the input token embeddings outputted from the $(n-1)^{th}$ Transformer block, $\mathbf{Z}^n_{dep}$ represent the output of n^{th} Transformer block, LN denotes the layer normalization [41], and F3N [3] consists of a soft composition (SC) [3] and a soft split operation. Note that F3N is adopted in our work to build up the interaction of overlapped token embeddings for effective feature propagation. We stack P spatio-temporal Transformer blocks and use them to implement feature propagation, enabling the module to effectively combine the cues collected from local and non-local depth to generate the complete depth.

3) Depth Construction: After feature propagation, the token embeddings are accordingly updated. In order to obtain the complete depth, the updated tokens have to be converted back into features. To achieve this goal, the soft composition is applied to convert tokens into the overlapped feature patches. These patches are then used to form the complete features as

$$\hat{\mathbf{E}}_{dep} = \mathrm{SC}(\mathbf{Z}^{N_1}_{dep}), \tag{4}$$

where $\mathbf{Z}^P_{dep}$ denotes the output of the P^{th} Transformer block and $\hat{\mathbf{E}}_{dep}$ is the composed complete features of depth. Then, the features $\hat{\mathbf{E}}_{dep}$ are decoded by a CNN-based decoder consisting of four convolutional layers to generate the predicted depth $\mathbf{D}_c$ which has the same resolution as the input frame.

C. Content Reconstruction

The content reconstruction module is designed to generate contents for the broken foreground and background of missing regions. Note that the depth indicates both the contour and content layer information for a picture. Given a picture whose missing content crosses foreground and background, introducing depth as guidance to reconstruct the missing content may produce result with clear shape and structure. In this work, we construct the content reconstruction module based on the depth-guided spatio-temporal Transformer and the proposed multi-head mutual-self-attention. With this module, we can combine the spatial and temporal dependencies of frames with the guidance of depth to produce reasonable and reliable content for the target region.

Our proposed content reconstruction module is also developed based on the CNN-Transformer architecture and consists

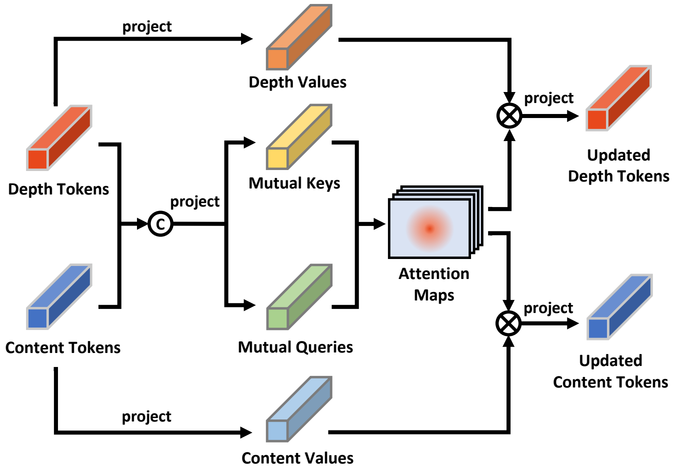

Fig. 4. Architecture of multi-head mutual self-attention mechanism (MMSA).

of feature extraction, depth-guided feature propagation and content composition. Specifically, given broken frames $\mathbf{X}_{in}$ and the corresponding complete depth maps $\mathbf{D}_c$, the lower-resolution features of content and depth are firstly extracted from the broken video frames and the predicted depth via two CNN-based encoders, respectively. Then, these features are converted into tokens and fed into the depth-guided STTB in which the tokens of content are updated through the depth-guided feature propagation. The updated tokens are converted into features and these features are finally used to reconstruct frames $\hat{\mathbf{X}}_{rec}$ via a CNN-based decoder.

1) Feature Extraction: Firstly, a context encoder [9] which consists of nine convolutional layers takes in the incomplete frames $\mathbf{X}_{in}$ and produces $1/4$ sized feature with C_{cn} channels. And the corresponding predicted depth $\mathbf{D}_c$ is feed into a CNN-based encoder with four convolutional layers to obtain $1/4$ sized feature maps $\mathbf{E}_{dp}$ with C_{dep} channels. Then, we apply the soft split to convert $\mathbf{E}_{cn}$ and $\mathbf{E}_{dp}$ into overlapped token embeddings as

$$\begin{aligned}\mathbf{Z}^0_{cn} &= \mathrm{SS}(\mathbf{E}_{cn})\\ \mathbf{Z}^0_{dp} &= \mathrm{SS}(\mathbf{E}_{dp}),\end{aligned} \tag{5}$$

where $\mathbf{Z}^0_{cn}$ and $\mathbf{Z}^0_{dp}$ represent the embedded tokens for $\mathbf{E}_{cn}$ and $\mathbf{E}_{dp}$, respectively.

2) Depth-guided Feature Propagation: Since the depth tokens contain enough depth information, we use them to guide the construction of spatio-temporal relationship between content tokens during feature propagation, which makes the model learn how to utilize depth index to assign appropriate attention weights for token updating. To achieve the above goal, we construct the depth-guided spatio-temporal Transformer block (DGSTTB) to facilitate the feature propagation. In addition, the multi-head mutual self attention (MMSA) is adopted in DGSTTB to make the content tokens update with the interaction of depth tokens.

To implement MMSA in DGSTTB, the depth tokens just participate in the assignment of attention weights but are not used to determine value vectors. Hence, the module can focus on the reference regions with similar depth to the target region and spread these cues to the target region so that it can effectively update token embeddings for missing regions to achieve better content reconstruction. Different from MSA, we use mutual query $\mathbf{Q}_{mul}$ and mutual key $\mathbf{K}_{mul}$ to obtain the attention map as well as the content value $\mathbf{V}_{cn}$ and the depth value $\mathbf{V}_{dp}$ for value updating in MMSA, as illustrated in Fig. 4, where $\mathbf{V}_{cn}$ and $\mathbf{V}_{dp}$ are independent. More specifically, we concatenate $\mathbf{Z}_{cn}$ with $\mathbf{Z}_{dp}$ together and use a linear projection layer f_{kq} to convert them into the mutual query vector $\mathbf{Q}_{mul}$ and mutual key vector $\mathbf{K}_{mul}$, respectively, i.e.,

$$\{\mathbf{K}^n_{mul}, \mathbf{Q}^n_{mul}\} = f_{kq}(\mathrm{Concat}(\mathbf{Z}^{n-1}_{cn}, \mathbf{Z}^{n-1}_{dp})), \tag{6}$$

where $\mathbf{Z}^{n-1}_{cn}$ and $\mathbf{Z}^{n-1}_{dep}$ represent the output content and depth token embeddings of the $(n-1)^{th}$ DGSTTB and $\mathbf{K}^n_{mul}$ and $\mathbf{Q}^n_{mul}$ are mutual key and query vectors of the n^{th} DGSTTB. In addition, $\mathbf{Z}_{cn}$ and $\mathbf{Z}_{dp}$ are independently converted into the content and depth values

$$\begin{aligned}\mathbf{V}^n_{cn} &= f_{vc}(\mathbf{Z}^{n-1}_{cn})\\ \mathbf{V}^n_{dp} &= f_{vd}(\mathbf{Z}^{n-1}_{dp}),\end{aligned} \tag{7}$$

where f_{vc} and f_{vd} are the linear projection layers used to generate the content and depth values, respectively. In order to capture various relationships between tokens, $\mathbf{Q}_{mul}$, $\mathbf{K}_{mul}$, $\mathbf{V}_{cn}$ and $\mathbf{V}_{dp}$ are then split into multiple heads. For each head, we generate its corresponding attention map by using mutual query and key. Then, we assign the weights for token updating according to the attention map.

Assuming that there are k heads in MMSA, the updated content and depth values for each head, denoted as $\hat{\mathbf{V}}_{cn,k}$ and $\hat{\mathbf{V}}_{dp,k}$, are obtained as

$$\begin{aligned}\{\hat{\mathbf{V}}_{cn,k}, \hat{\mathbf{V}}_{dp,k}\} =&\ \mathrm{MMSA}(\mathbf{Q}_{mul,k}, \mathbf{K}_{mul,k}, \mathbf{V}_{cn,k}, \mathbf{V}_{dp,k})\\ =&\ \frac{softmax(\mathbf{Q}_{mul,k}\mathbf{K}^T_{mul,k})}{\sqrt{d_k}}\{\mathbf{V}_{cn,k}, \mathbf{V}_{dp,k}\},\end{aligned} \tag{8}$$

where $\mathbf{Q}_{mul,k}$, $\mathbf{K}_{mul,k}$, $\mathbf{V}_{cn,k}$, $\mathbf{V}_{dp,k}$ are the mutual query, mutual key, content value and depth value for each head, respectively, and d_k is the feature dimension of $\mathbf{Q}_{mul,k}$.

Based on MMSA, DGSTTB is implemented as

$$\begin{aligned}\{\mathbf{Z}'^n_{cn}, \mathbf{Z}'^n_{dp}\} =&\ \mathrm{MMSA}(\{\mathrm{LN}(\mathbf{Z}^{n-1}_{cn}), \mathrm{LN}(\mathbf{Z}^{n-1}_{dp})\})\\ &+ \{\mathbf{Z}^{n-1}_{cn}, \mathbf{Z}^{n-1}_{dp}\}\\ \mathbf{Z}^n_{cn} =&\ \mathrm{F3N}(\mathrm{LN}(\mathbf{Z}'^n_{c})) + \mathbf{Z}'^n_{cn}\\ \mathbf{Z}^n_{dp} =&\ \mathrm{F3N}(\mathrm{LN}(\mathbf{Z}'^n_{dp})) + \mathbf{Z}'^n_{dp},\end{aligned} \tag{9}$$

where $\mathbf{Z}^{n-1}_{dp}$ denotes the output token embeddings of the $(n-1)^{th}$ DGSTTB and $\mathbf{Z}^n_{dp}$ represents the output of the n^{th} transformer block. We stack Q DGSTTBs and use them to facilitate the depth-guided feature propagation, enabling the model to effectively update the tokens for broken regions to produce reliable contents for missing regions.

3) Initial Frame Composition: After the feature propagation, the content tokens are accordingly updated. In order to reconstruct frames, the overlapped token embeddings for contents are converted into features by the CNN-based decoder with the SC operation as

$$\hat{\mathbf{E}}_{rec} = \mathrm{SC}(\mathbf{Z}_{cn}^{N_2}), \tag{10}$$

where $\mathbf{Z}_{cn}^{Q}$ denotes the content tokens obtained from the Q^{th} DGSTTB and $\hat{\mathbf{E}}_{rec}$ is the composed features obtained by using SC. Note that the features $\hat{\mathbf{E}}_{rec}$ only contain the features of local frames and the features of non-local frames are discarded. Then we apply the CNN-based decoder which consists of four convolutional layers to progressively upsample $\hat{\mathbf{E}}_{rec}$ and generate initial results for local frames. After that, the composed local frames $\hat{\mathbf{X}}_{rec}$ and the composed features of local frames $\hat{\mathbf{E}}_{rec}$ are fed into the content enhancement module to generate the final inpainting results.

D. Content Enhancement

After composing the frames for the broken video, we further improve the video quality by introducing optical flow as the guidance to enhance the temporal coherence of neighboring frames. The content enhancement module is accordingly designed to enhance the local temporal consistency and the texture quality for the video. This module is constructed based on a parallel feature fusion network with the flow-guided deformable warping. With this module, we can strengthen the local temporal coherence of the video to enhance the visual consistency and improve both structure and texture details for the final inpainting result.

In this work, we develop a parallel content enhancement module based on the flow-guided deformable convolution [42] to facilitate the content enhancement that consists of flow estimation, feature enhancement and final frame reconstruction. Specifically, given the composed local frames $\hat{\mathbf{X}}_{rec}$, we predict forward and backward flows for them with a flow estimation network. Then, with the features $\hat{\mathbf{E}}_{rec}$ obtained from content reconstruction module, we implement flow-guided deformable warping to simultaneously warp the features of the neighboring frames to the target frame and fuse the warped features for neighboring frames with target feature maps using multi-layer perception (MLP) to enhance the features of target frame. After that, we feed the enhanced features into a CNN-based decoder to generate the final inpainted frames $\bar{\mathbf{X}}_{en}$.

1) Flow Estimation: We adopt a lightweight flow estimation network SpyNet [43] to predict the forward and backward flows between neighboring frames so as to save the computation cost. We use $F_{t-1\rightarrow t}$ and $F_{t+1\rightarrow t}$ to denote the forward and backward flows, respectively.

2) Feature Enhancement: As illustrated in Fig. 5, we construct a parallel content enhancement block (PCEB) that is developed based on flow-guided deformable warping to strengthen the temporal coherence of the video. In this work, the proposed PCEB is applied to enhance a group of local neighboring frames $\{..., \hat{X}_{t-3}, \hat{X}_{t-2}, \hat{X}_{t-1}, \hat{X}_{t+1}, \hat{X}_{t+2}, \hat{X}_{t+3}, ...\}$. We describe its implementation using two neighboring frames, $\{\hat{X}_{t-1}, \hat{X}_{t+1}\}$, but it can also be extended to work with four frames, $\{\hat{X}_{t-2}, \hat{X}_{t-1}, \hat{X}_{t+1}, \hat{X}_{t+2}\}$, or six frames, $\{\hat{X}_{t-3}, \hat{X}_{t-2}, \hat{X}_{t-1}, \hat{X}_{t+1}, \hat{X}_{t+2}, \hat{X}_{t+3}\}$, in the implementation of our method. Given the features $\hat{\mathbf{E}}_l$ of the composed frames $\hat{\mathbf{X}}_l$, we stack K PCEBs to produce the enhanced features $\bar{\mathbf{E}}_{en}$ and each PCEB is implemented in four steps.

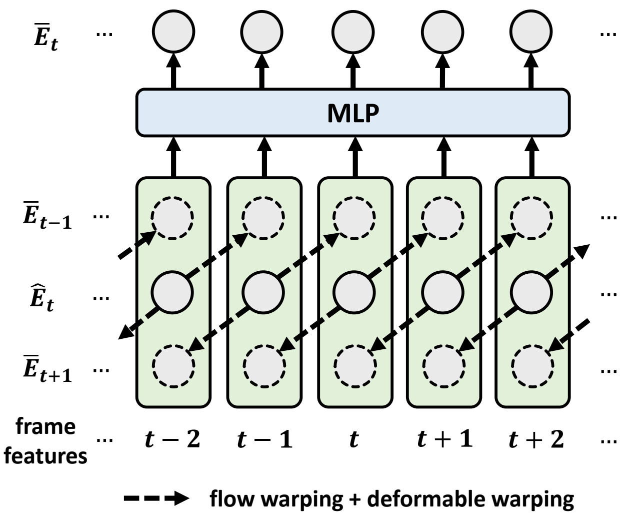

Fig. 5. Architecture of parallel content enhancement block (PCEB).

Firstly, the features $\hat{\mathbf{E}}_{t-1}$ and $\hat{\mathbf{E}}_{t+1}$ extracted from the frames $\hat{X}_{t-1}$ and $\hat{X}_{t+1}$ are simultaneously warped to the features $\hat{\mathbf{E}}_t$ of the inpainted frame $\hat{X}_t$ as

$$\begin{aligned} \hat{\mathbf{E}}_{t-1}^{'} &= \mathcal{W}(\hat{\mathbf{E}}_{t-1}, \mathbf{F}_{t-1\rightarrow t}) \\ \hat{\mathbf{E}}_{t+1}^{'} &= \mathcal{W}(\hat{\mathbf{E}}_{t+1}, \mathbf{F}_{t+1\rightarrow t}), \end{aligned} \tag{11}$$

where $\mathcal{W}(\cdot)$ denotes the flow-based warping [42], [44].

Secondly, we predict the offset residuals and modulation masks with several convolution layers so that we can use them to implement the deformable warping for feature enhancement. The offset residuals and modulation masks are obtained as

$$\begin{aligned} &\{\mathbf{O}_{t-1\rightarrow t}, \mathbf{O}_{t+1\rightarrow t}, \mathbf{M}_{t-1\rightarrow t}, \mathbf{M}_{t+1\rightarrow t}\} \\ &= \mathrm{Conv}(\mathrm{Concat}(\mathbf{F}_{t-1\rightarrow t}, \mathbf{F}_{t+1\rightarrow t}, \hat{\mathbf{E}}_{t-1}^{'}\hat{\mathbf{E}}_{t+1}^{'})), \end{aligned} \tag{12}$$

where $\mathbf{O}_{t-1\rightarrow t}$ and $\mathbf{O}_{t+1\rightarrow t}$ are the predicted offset residuals, $\mathbf{M}_{t-1\rightarrow t}$ and $\mathbf{M}_{t+1\rightarrow t}$ are the predicted modulation masks, and Conv denotes the application of convolutional layers.

Thirdly, with the predicted offset residuals and modulation masks, we employ the deformable warping to warp features $\bar{\mathbf{E}}_{t-1}$ and $\bar{\mathbf{E}}_{t+1}$ to guarantee the features of neighboring frames can be effectively aligned to $\hat{\mathbf{E}}_t$. The deformable warping is implemented as

$$\begin{aligned} \bar{\mathbf{E}}_{t-1} &= \mathrm{DConv}(\hat{\mathbf{E}}_{t-1}^{'}, \mathbf{F}_{t-1\rightarrow t} + \mathbf{O}_{t-1\rightarrow t}, \mathbf{M}_{t-1\rightarrow t}) \\ \bar{\mathbf{E}}_{t+1} &= \mathrm{DConv}(\hat{\mathbf{E}}_{t+1}^{'}, \mathbf{F}_{t+1\rightarrow t} + \mathbf{O}_{t+1\rightarrow t}, \mathbf{M}_{t+1\rightarrow t}), \end{aligned} \tag{13}$$

where DConv denotes the deformable convolution [45].

Finally, we concatenate $\bar{\mathbf{E}}_{t-1}$, $\bar{\mathbf{E}}_{t+1}$ and $\hat{\mathbf{E}}_t$ to fuse them with MLP to obtain the enhanced features $\bar{\mathbf{E}}_t$ as

$$\bar{\mathbf{E}}_t = \mathrm{MLP}(\mathrm{Concat}(\hat{\mathbf{E}}_t, \bar{\mathbf{E}}_{t-1}, \bar{\mathbf{E}}_{t+1})), \tag{14}$$

where Concat is the concatenation operation. The features of each frame can be simultaneously fused with the features of its neighboring frames to obtain all the enhanced features $\bar{\mathbf{E}}_{en}$.

3) Enhanced Frame Reconstruction: After obtaining the enhanced features $\bar{\mathbf{E}}_{en}$ from the K^{th} PCEB, we feed them into a CNN-based decoder that consists of four convolutional layers to progressively increase the resolution of features and generate the enhanced inpainting frames $\bar{\mathbf{X}}_{en}$. All the enhanced frames are used to compose the final output video.

E. Loss Function

We construct the loss function $\mathcal{L}_{total}$ to train our model and jointly optimizing all the modules. $\mathcal{L}_{total}$ is composed as

$$\begin{aligned}\mathcal{L}_{total} =&\lambda_{dep} \cdot \mathcal{L}_{dep} + \lambda_{con} \cdot \mathcal{L}_{con} + \lambda_{enh} \cdot \mathcal{L}_{enh} \\ &+ \lambda_{gen} \cdot \mathcal{L}_{gen},\end{aligned} \tag{15}$$

where $\mathcal{L}_{dep}$ is the depth completion loss, $\mathcal{L}_{con}$ is the content construction loss, $\mathcal{L}_{enh}$ is the content enhancement loss, $\mathcal{L}_{gen}$ is the T-PatchGAN loss [31], and λ_{dep}, λ_{con}, λ_{enh} and λ_{gen} are the corresponding weighting factors for each loss.

In $\mathcal{L}_{total}$, the depth completion loss measures the difference between the predicted depth information $\hat{\mathbf{D}}$ and the ground-truth depth information $\mathbf{D}$. It is defined as

$$\mathcal{L}_{dep} = \|\hat{\mathbf{D}} - \mathbf{D}\|_1. \tag{16}$$

The content construction loss $\mathcal{L}_{con}$ measures the difference between the reconstructed video $\hat{\mathbf{X}}$ obtained from the content reconstruction module and the ground-truth video $\mathbf{X}$. It is formulated as

$$\mathcal{L}_{con} = \|\hat{\mathbf{X}} - \mathbf{X}\|_1. \tag{17}$$

The content enhancement loss $\mathcal{L}_{enh}$ measures the difference between the final output video $\bar{\mathbf{X}}$ obtained from the content enhancement module and the ground-truth video $\mathbf{X}$, i.e.,

$$\mathcal{L}_{enh} = \|\bar{\mathbf{X}} - \mathbf{X}\|_1. \tag{18}$$

The T-PatchGAN loss [31] evaluates the difference between the final inpainted video $\bar{\mathbf{X}}$ and the ground-truth video $\mathbf{X}$ with a T-PatchGAN discriminator [31] $\mathcal{D}$, where the discriminator makes the model generate high-quality and realistic contents. The T-PatchGAN loss is formulated as

$$\mathcal{L}_{gen} = -\mathbb{E}_{\bar{\mathbf{X}}}[\mathcal{D}(\bar{\mathbf{X}})]. \tag{19}$$

Moreover, the T-PatchGAN discriminator consists of six 3D convolution layers and is used to learn the difference between real patches of ground-truth videos and fake patches of inpainted videos. The loss adopted in Chang's work [31] is employed to train the discriminator in this work, making the discriminator correctly classify real and fake samples with a clear margin. It is formulated as

$$\begin{aligned}\mathcal{L}_D =&E_{x\sim P_{\mathbf{X}}(x)}[\max(0, 1-\mathcal{D}(x))] \\ &+ E_{z\sim P_{\bar{\mathbf{X}}}(z)}[\max(0, 1+\mathcal{D}(z))],\end{aligned} \tag{20}$$

where $\mathcal{D}(x)$ represents the discriminator's output for a real video sample x and $\mathcal{D}(z)$ represents the output for an inpainting video sample z.

IV. Experimental Results

A. Settings

1) Datasets: We evaluate the proposed method on two widely used video object segmentation datasets, YouTube-VOS [46] and DAVIS [47], to demonstrate its effectiveness. The YouTube-VOS dataset consists of 3,471, 474, and 508 video clips for training, validation, and testing, respectively, covering various scenes. The DAVIS dataset contains 60 videos in the training set and 90 videos in the test set.

We train our model using the YouTube-VOS dataset and evaluate the performance using both the DAVIS and YouTube-VOS datasets. Specifically, following the initial partitioning of YouTube-VOS dataset, we use its training set to train our model. Moreover, to make our proposed method applicable to different inpainting scenarios, we create both the stationary irregular masks and the dynamic object-shaped masks as [2]–[4], [28], [30], [32] did, and apply them to the source videos to produce broken videos by removing the masked contents. To evaluate the performance of the method, we conduct evaluations on the YouTube-VOS test set and 50 video clips from the test set of DAVIS dataset as the previous work [2]–[4] did.

2) Implementation Details: During training, the numbers of local frames N_l and non-local frames N_{nl} are both set to 4. During test, the number of local frames N_l is set to 6, while the step-size to uniformly sample non-local frames N_{nl} is set to 6. In the experiment, the model adopt 8 STTBs in the depth completion module, 8 DGSTTBs in the content reconstruction module and 4 PCEBs in the content enhancement module, i.e., $P = 8$, $Q = 8$, and $K = 4$. The number of content feature C_{cn} is set to 128, while the number of depth feature C_{dep} is set to 64 in the depth completion and 32 in the content reconstruction. The head number k of both MSA and MMSA are set to 4. We first train the depth completion module independently using $\mathcal{L}_{dep}$ for $300K$ iterations. Then with the depth completion module and the pretrained flow estimation network, SpyNet, frozen, we train the content reconstruction and content enhancement modules using $\mathcal{L}_{con}$, $\mathcal{L}_{enh}$ and $\mathcal{L}_{gen}$ for $300K$ iterations. And we finetune three modules together using $\mathcal{L}_{total}$ for $200K$ iteration. The weighting factors for λ_{dep}, λ_{con}, λ_{enh} and λ_{gen} are set to 0.2, 0.2, 1 and $1e{-}3$, respectively. We adopt Adam optimizer [48] to train our network. The initial learning rate is $1e{-}4$, which is divided by 10 after $150K$ iterations. The resolution of training videos are resized to 240×432 and the batch size is set to 4. The training of our network is implemented on Pytorch platform with two NVIDIA GeForce RTX 3090 GPUs, while the test experiments are implemented with one NVIDIA GeForce RTX 3090 GPU.

3) Evaluation Metrics: We adopt peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) [49], video Frechet inception distance (VFID) [50], and flow warping error (E_{warp}) [51] as the quantitative metrics to evaluate the performance of different video inpainting methods. More specifically, PSNR and SSIM are two widely used metrics to assess reconstructed image and video with original ones. Higher value suggests higher similarity. VFID is employed to assess

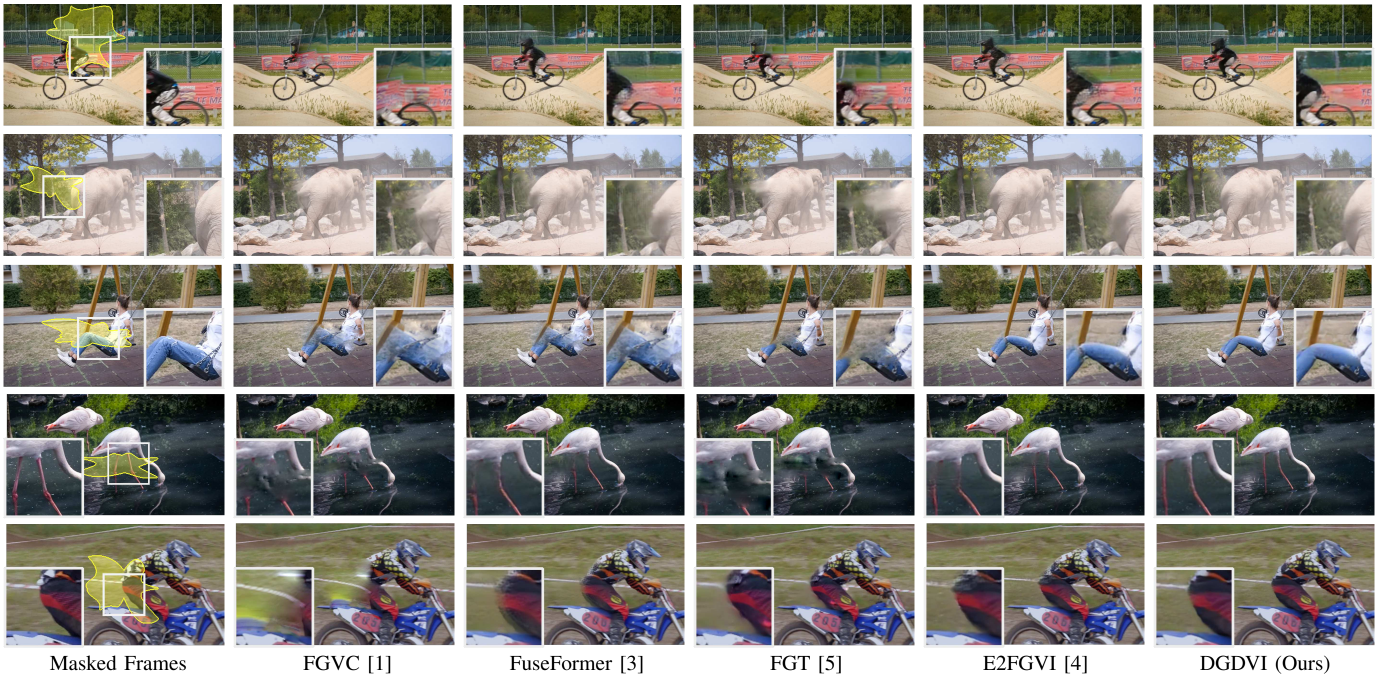

Fig. 6. Qualitative results for the video completion scenario that crosses foreground and background. From top to bottom: *Bmx-bumps*, *Elephant*, *Swing*, *Flamingo*, and *Motocross-bumps* videos of the DAVIS dataset.

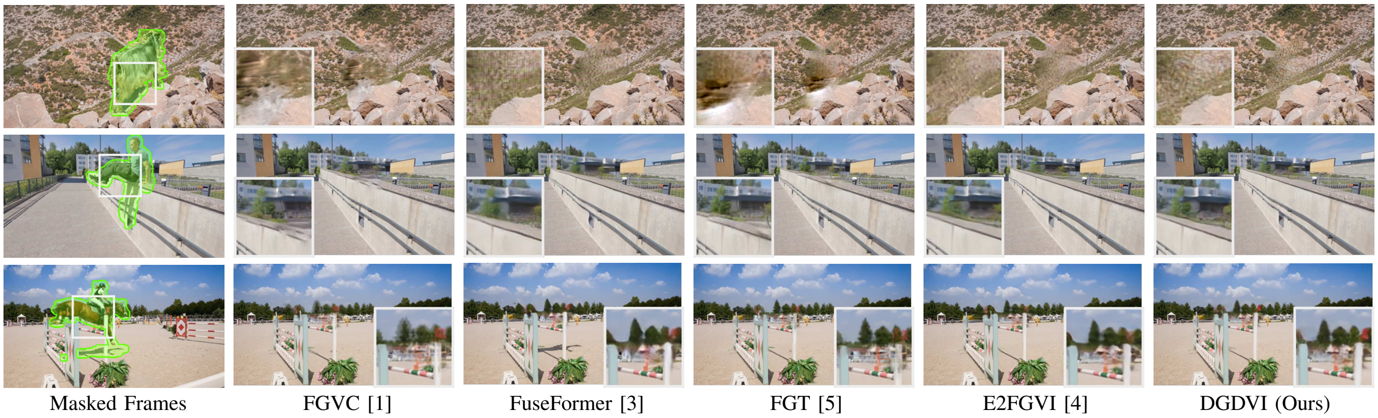

Fig. 7. Qualitative results for the object removal scenario. From top to bottom: *Goat*, *Parkour*, and *Horsejump-high* videos of the DAVIS dataset.

the perceptual similarity of distortion-oriented videos and has been adopted in recent video inpainting approaches [2]–[4]. Lower value represents better realism and less distortion compared with natural videos. Flow warping error E_{warp} measures the temporal consistency based on optical flow. Lower score indicates better temporal consistency.

B. Comparisons

1) Qualitative Results: We qualitatively compare our method with four latest approaches, including flow-edge guided video completion (FGVC) [1], FuseFormer [3], flow-guided Transformer for video inpainting (FGT) [5] and end-to-end framework for flow-guided video inpainting (E2FGVI) [4].

The comparison is conducted on two tasks. The first one is video completion and the second one is object removal, where both the tasks are performed on the videos of the DAVIS dataset. Moreover, we create the stationary irregular mask for the video completion task as the previous methods [2]–[4], [38] did. In this task, one static mask is randomly applied to a video to remove the content. In contrast, we produce dynamic object-shaped masks for the object removal task, where each mask covers one moving object over the whole video.

Some video completion results for the challenging scenes crossing foreground and background are presented in Fig. 6 and some object removal results are presented in Fig. 7. One can see from Fig. 6 and Fig. 7 that our method generates more reliable contents and clearer structures than the other approaches, demonstrating its effectiveness.

2) Quantitative Results: We conduct quantitative comparison on YouTube-VOS and DAVIS for video completion. The resolution of test videos is 240×432. The proposed method is compared to deep video inpainting (VINet) [30], deep flow-guided video inpainting (DFVI) [6], learnable gated temporal shift module (LGTSM) [32], copy-and-paste networks (CAP) [28], spatial-temporal transformations for video inpainting (STTN) [2], axial attention-based style Transformer

TABLE I
QUANTITATIVE COMPARISONS ON YOUTUBE-VOS [46] AND DAVIS [47] DATASETS. ↑ INDICATES HIGHER IS BETTER. ↓ INDICATES LOWER IS BETTER. $E_{warp}{}^{*}$ DENOTES $E_{warp} \times 10^{-2}$. EACH METHOD IS EVALUATED FOLLOWING THE PROCEDURES IN FUSEFORMER. VINET, DFVI, FGVC, AND FGT ARE NOT END-TO-END TRAINING METHODS. THEIR FLOPS, THUS, ARE NOT PRESENTED. AAST DID NOT PROVIDE THE SOURCE CODE. AS SUCH ITS FLOPS AND RUNTIME ARE NOT PROVIDED.

	Accuracy								Efficiency	
	YouTube-VOS				DAVIS				FLOPs	Runtime (s/frame)
Models	PSNR (dB) ↑	SSIM ↑	VFID ↓	$E_{warp}{}^{*}$ ↓	PSNR (dB) ↑	SSIM ↑	VFID ↓	$E_{warp}{}^{*}$ ↓		
VINet [30]	29.20	0.9434	0.072	0.1490	28.96	0.9411	0.199	0.1785	-	-
DFVI [6]	29.16	0.9429	0.066	0.1509	28.81	0.9404	0.187	0.1608	-	2.56
LGTSM [32]	29.74	0.9504	0.070	0.1859	28.57	0.9409	0.170	0.1640	1008G	0.23
CAP [28]	31.58	0.9607	0.071	0.1470	30.28	0.9521	0.182	0.1533	861G	0.40
FGVC [1]	29.67	0.9403	0.064	0.1022	30.80	0.9497	0.165	0.1586	-	2.36
STTN [2]	32.34	0.9655	0.053	0.0907	30.67	0.9560	0.149	0.1449	1032G	0.12
FuseFormer [3]	33.29	0.9681	0.053	0.0900	32.54	0.9700	0.138	0.1362	752G	0.20
AAST [38]	33.23	0.9669	0.048	0.1396	32.71	0.9720	0.1360	0.1706	-	-
FGT [5]	30.19	0.9536	0.063	0.0968	31.77	0.9639	0.134	0.1483	-	1.89
E2FGVI [4]	33.71	0.9700	0.046	0.0864	33.01	0.9721	0.116	0.1315	682G	0.16
DGDVI (Ours)	**34.07**	**0.9725**	**0.045**	**0.0823**	**33.33**	**0.9740**	**0.111**	**0.1295**	860G	0.21

TABLE II
ABLATION STUDY FOR THE PROPOSED MODULES

	Model-1	*Model*-2	*Model*-3	*Model*-4
Content reconstruction	✓	✓	✓	✓
+ Depth completion	✗	✗	✓	✓
+ Content enhancement	✗	✓	✗	✓
PSNR (dB) / SSIM	32.54/0.9700	33.04/0.9718	33.08/0.9724	**33.33/0.9740**

TABLE III
ABLATION STUDY FOR MMSA

	PSNR (dB) ↑	SSIM ↑	VFID ↓
MSA	32.74	0.9716	0.124
MMSA	**33.30**	**0.9740**	**0.111**

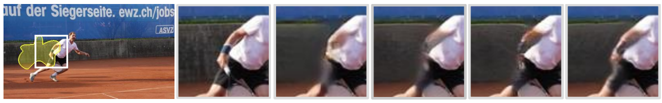

Fig. 8. Ablation study for the proposed modules. From left to right: Masked frame of *Tennis* video, portions for ground truth, model-1, model-2, model-3, and model-4.

TABLE IV
ABLATION STUDY FOR FLOW-GUIDED DEFORMABLE WARPING

	PSNR (dB) ↑	SSIM ↑	VFID ↓
Flow-based	32.75	0.9710	0.121
Deformable	32.61	0.9691	0.125
Flow-guided deformable	**33.30**	**0.9740**	**0.111**

(AAST) [38], FGVC [1], FuseFormer [3], FGT [5], and E2FGVI [4]. The corresponding results are given in Table I. It is found from Table I that our method significantly outperforms all the state-of-the-art methods evaluated by the four quantitative metrics. These results indicate that our approach can recover the contents with less distortion (PSNR and SSIM), more visually faithful content (VFID), and better spatial and temporal consistency (E_{warp}).

3) Complexity: We use floating point operations (FLOPs) and inference time to evaluate the complexity of the compared methods by using the DAVIS dataset. The corresponding results are presented in Table I. The FLOPs of our proposed approach are comparable to VINet [30], LGTSM [32] and CAP [28] that are developed based on CNN. Meanwhile, the proposed method executes about $\times 10$ faster than DFVI [6], FGVC [1] and FGT [5]. In these methods, the optical flow is adopted to guide the information propagation throughout the frames for inpainting, resulting in rather high complexity. Meanwhile, our method achieves comparable speeds to the Transformer-based approaches, such as STTN [2], FuseFormer [3], and E2FGVI [4].

C. Ablation Study

We conduct ablation studies to verify the effectiveness of the proposed modules, MMSA and flow-guided deformable warping used in our model. All the studies are performed on the DAVIS dataset for the video completion task.

1) Effectiveness of the proposed modules: Our proposed inpainting model consists of three modules, i.e., the depth completion, content reconstruction, and content enhancement modules. To demonstrate the performance gain offered by them, we conduct an ablation study to verify their effectiveness. The content reconstruction module is the key module in our model. Once it is removed, our proposed inpainting model will not work any longer. Therefore, it is always retained in our model when the ablation study was carried out.

When we conduct this ablation study, we firstly just use the content reconstruction module to construct a baseline model, denoted as *Model-1*, where the content reconstruction module is implemented with MSA rather than MMSA (as depth was not available for guidance). Then, we add the content enhancement module to *Model-1* to compose *Model-2* for the verification of effectiveness of this module. Meanwhile, we compose *Model-3* by using the content reconstruction and depth completion modules, where MMSA is adopted in content reconstruction because the depth can be offered by the depth completion module. Finally, we integrate all the modules to build up our proposed inpainting model (denoted as *Model-4* in this experiment). The quantitative and qualitative results for the ablation study are given in Table II and Fig. 8, respectively.

According to the results presented in Table II and Fig. 8,

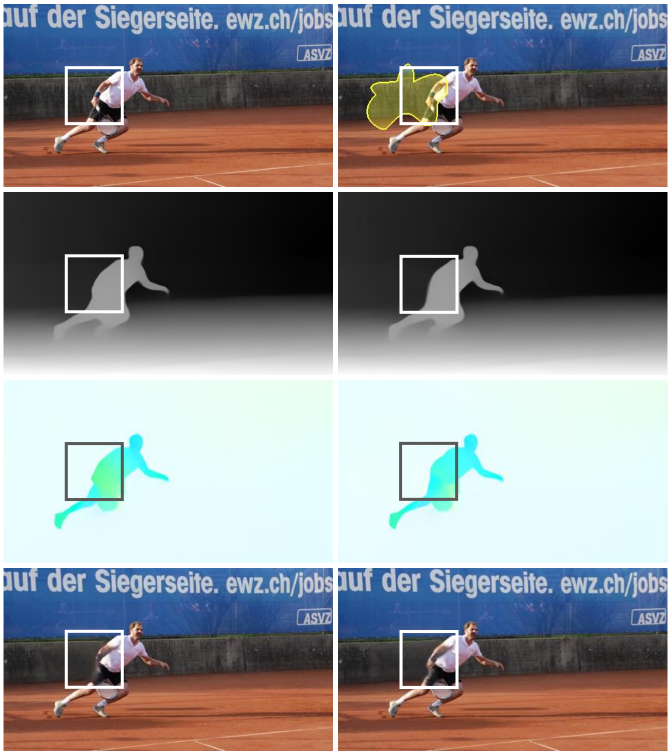

Fig. 9. The intermediate results for DGDVI. The example is selected the same frame as Fig. 8 from *Tennis* video. From left to right, top to bottom: Ground-truth frame, mask, depth estimation result for ground-truth frame, depth completion result for broken video, visualization of flow estimation result for ground-truth video, visualization of flow estimation for initial inpainted video, initial inpainted result, enhanced inpainted result.

TABLE V
QUANTITATIVE RESULTS FOR THE EFFECTIVENESS VERIFICATION OF ADOPTING NON-LOCAL FRAMES TO IMPLEMENT INPAINTING

	PSNR (dB) ↑	SSIM ↑	VFID ↓
Local frames	32.55	0.9687	0.126
Local + non-local frames	**33.30**	**0.9740**	**0.111**

it is found that introducing the depth completion and content enhancement modules to the baseline model, i.e., *Model-1*, can effectively improve the inpainting quality, both quantitatively and qualitatively, When all the modules are adopted, the best quality is achieved. These results demonstrate the effectiveness of the proposed modules.

In addition, we present some visualized results in Fig. 9 to further verify the effectiveness of our proposed modules. Firstly, it is found from Fig. 9 that the predicted depth for the broken video is very similar to the depth of the ground-truth video. This accordingly demonstrates the effectiveness of the depth completion module. Secondly, guided by the predicted depth, we obtained the initial inpainted video with acceptable quality by using the content reconstruction module. Note that employing this initial result can generate optical flow similar to the one obtained from the ground-truth video. Finally, with the obtained flow, we enhance the initial result and get the final inpainting result with higher quality, which validates the effectiveness of the content enhancement module.

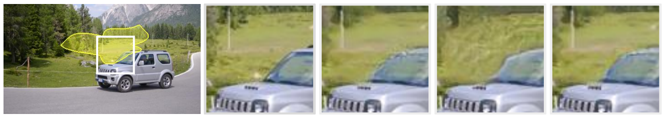

Fig. 10. Ablation study for flow-guided deformable warping. From left to right: Masked frame of *Car-turn* video, portions for ground truth, flow-based warping, deformable convolution-based warping, and flow-guided deformable warping.

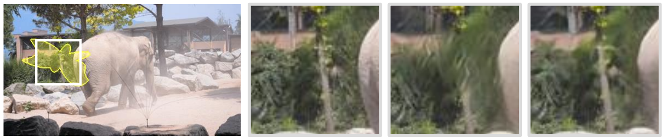

Fig. 11. Qualitative results for the effectiveness verification of adopting non-local frames to implement inpainting. From left to right: Masked frame of *Elephant* video, portions for ground truth, inpainting just with local frames, and inpainting with both local and non-local frames.

2) Effectiveness of MMSA for content reconstruction: The depth-guided feature propagation in the content reconstruction is developed based on the proposed MMSA mechanism. To verify the effectiveness of MMSA, we replace it with MSA in content reconstruction to evaluate the change of inpainting performance. Specifically, the content reconstruction with MSA first fuses depth and feature, and then feed them into STTBs for feature propagation. The quantitative results are given in Table III. According to the results in Table III, it is found that the model with MMSA achieves better inpainting performance than using MSA, which demonstrates the superiority of MMSA for the content reconstruction.

3) Effectiveness of the flow-guided deformable warping for content enhancement: The content enhancement module is developed based on the flow-guided deformable warping. In order to verify the superiority of this warping approach, we compare its performance with two warping methods, flow-based warping and deformable convolution-based warping. Specifically, instead of using flow-guided deformable warping in content enhancement, we implement flow-based or deformable convolution-based warping to align features for feature enhancement. The corresponding quantitative and qualitative results inpainting results are presented in Table IV and Fig. 10, respectively. It can be found from Table IV and Fig. 10 that the model with flow-guided deformable warping generated the best results, demonstrating the effectiveness of this warping technique.

4) Effectiveness of adopting non-local frames for video inpainting: To validate the effectiveness of adopting non-local frames in the inpainting, we firstly conduct an experiment by just employing the local frames to implement the inpainting with our proposed model. Then, we compare the inpainting results with the ones obtained by employing both local and non-local frames to inpaint videos. The corresponding quantitative and qualitative results are given in Table V and Fig. 11, respectively. These results demonstrate the superior performance of our approach when both local and non-local frames are adopted.

V. Conclusion

In this paper, we propose a depth-guided deep inpainting network for videos. Our proposed model is composed of three integral modules: depth completion, content reconstruction, and content enhancement, implemented in a sequential workflow. More specifically, the depth completion module is developed based on the spatio-temporal Transformer and used to obtain the completed depth information for video frame. The content reconstruction module is constructed to obtain the initially inpaint video with the guidance of depth information. The content enhancement module is developed to enhance the quality of the video. These modules are jointly optimized so as to guarantee the high inpainting efficiency. The experimental results demonstrate that our proposed method offers better inpainting results compared with the state-of-the-art methods.

References

[1] C. Gao, A. Saraf, J.-B. Huang, and J. Kopf, “Flow-edge guided video completion,” in *Proc. Eur. Conf. Comput. Vis.*, 2020.

[2] Y. Zeng, J. Fu, and H. Chao, “Learning joint spatial-temporal transformations for video inpainting,” in *Proc. Eur. Conf. Comput. Vis.*, 2020.

[3] R. Liu, H. Deng, Y. Huang, X. Shi, L. Lu, W. Sun, X. Wang, J. Dai, and H. Li, “Fuseformer: Fusing fine-grained information in transformers for video inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021.

[4] Z. Li, C.-Z. Lu, J. Qin, C.-L. Guo, and M.-M. Cheng, “Towards an end-to-end framework for flow-guided video inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 17 562–17 571.

[5] K. Zhang, J. Fu, and D. Liu, “Flow-guided transformer for video inpainting,” in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 74–90.

[6] R. Xu, X. Li, B. Zhou, and C. C. Loy, “Deep flow-guided video inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2019.

[7] Y. Zhou, C. Barnes, E. Shechtman, and S. Amirghodsi, “Transfill: Reference-guided image inpainting by merging multiple color and spatial transformations,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2266–2276.

[8] M. Liao, F. Lu, D. Zhou, S. Zhang, W. Li, and R. Yang, “Dvi: Depth guided video inpainting for autonomous driving,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 1–17.

[9] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, “Context encoders: Feature learning by inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2536–2544.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[11] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, “Generative image inpainting with contextual attention,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5505–5514.

[12] H. Liu, B. Jiang, Y. Xiao, and C. Yang, “Coherent semantic attention for image inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 4170–4179.

[13] C. Xie, S. Liu, C. Li, M.-M. Cheng, W. Zuo, X. Liu, S. Wen, and E. Ding, “Image inpainting with learnable bidirectional attention maps,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8858–8867.

[14] Z. Yi, Q. Tang, S. Azizi, D. Jang, and Z. Xu, “Contextual residual aggregation for ultra high-resolution image inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7508–7517.

[15] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, “Image inpainting for irregular holes using partial convolutions,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 85–100.

[16] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, “Free-form image inpainting with gated convolution,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 4471–4480.

[17] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky, “Resolution-robust large mask inpainting with fourier convolutions,” in *Proc. IEEE Winter Conf. App. Comput. Vis.*, 2022, pp. 2149–2159.

[18] K. Nazeri, E. Ng, T. Joseph, F. Z. Qureshi, and M. Ebrahimi, “Edgeconnect: Generative image inpainting with adversarial edge learning,” *arXiv preprint arXiv:1901.00212*, 2019.

[19] W. Xiong, J. Yu, Z. Lin, J. Yang, X. Lu, C. Barnes, and J. Luo, “Foreground-aware image inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5840–5848.

[20] Y. Ren, X. Yu, R. Zhang, T. H. Li, S. Liu, and G. Li, “Structureflow: Image inpainting via structure-aware appearance flow,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 181–190.

[21] H. Wu, J. Zhou, and Y. Li, “Deep generative model for image inpainting with local binary pattern learning and spatial attention,” *IEEE Trans. Multimedia*, vol. 24, pp. 4016–4027, 2021.

[22] Y. Song, C. Yang, Y. Shen, P. Wang, Q. Huang, and C.-C. J. Kuo, “Spg-net: Segmentation prediction and guidance network for image inpainting,” *arXiv preprint arXiv:1805.03356*, 2018.

[23] Y. Zhang, Y. Liu, R. Hu, Q. Wu, and J. Zhang, “Mutual dual-task generator with adaptive attention fusion for image inpainting,” *IEEE Trans. Multimedia*, 2023.

[24] H. Sun, W. Li, Y. Duan, J. Zhou, and J. Lu, “Learning adaptive patch generators for mask-robust image inpainting,” *IEEE Trans. Multimedia*, 2022.

[25] Y. Yu, F. Zhan, R. Wu, J. Pan, K. Cui, S. Lu, F. Ma, X. Xie, and C. Miao, “Diverse image inpainting with bidirectional and autoregressive transformers,” in *ACM Int. Conf. Multimedia*, 2021, pp. 69–78.

[26] W. Li, Z. Lin, K. Zhou, L. Qi, Y. Wang, and J. Jia, “Mat: Mask-aware transformer for large hole image inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10 758–10 768.

[27] Q. Dong, C. Cao, and Y. Fu, “Incremental transformer structure enhanced image inpainting with masking positional encoding,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11 358–11 368.

[28] S. Lee, S. W. Oh, D. Won, and S. J. Kim, “Copy-and-paste networks for deep video inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019.

[29] C. Wang, H. Huang, X. Han, and J. Wang, “Video inpainting by jointly learning temporal structure and spatial details,” in *Proc. AAAI Conf. Artif. Intell.*, 2019.

[30] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, “Deep video inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2019.

[31] Y. Chang, Z. Y. Liu, K. Lee, and W. Hsu, “Free-form video inpainting with 3d gated convolution and temporal patchgan,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019.

[32] Y.-L. Chang, Z. Y. Liu, K.-Y. Lee, and W. Hsu, “Learnable gated temporal shift module for deep video inpainting,” in *Proc. Brit. Mach. Vis. Conf.*, 2019.

[33] X. Zou, L. Yang, D. Liu, and Y. J. Lee, “Progressive temporal feature alignment network for video inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2021.

[34] K. Zhang, J. Fu, and D. Liu, “Inertia-guided flow completion and style fusion for video inpainting,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5982–5991.

[35] J. Kang, S. W. Oh, and S. J. Kim, “Error compensation framework for flow-guided video inpainting,” in *Proc. Eur. Conf. Comput. Vis.* Springer, 2022, pp. 375–390.

[36] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.

[37] R. Liu, H. Deng, Y. Huang, X. Shi, L. Lu, W. Sun, X. Wang, and L. Hongsheng, “Decoupled spatial-temporal transformer for video inpainting,” *arXiv preprint arXiv:2104.06637*, 2021.

[38] M. S. Junayed and M. B. Islam, “Consistent video inpainting using axial attention-based style transformer,” *IEEE Trans. Multimedia*, 2022.

[39] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, “Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[41] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.

[42] K. C. Chan, S. Zhou, X. Xu, and C. C. Loy, “Basicvsr++: Improving video super-resolution with enhanced propagation and alignment,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2022.

[43] A. Ranjan and M. J. Black, “Optical flow estimation using a spatial pyramid network,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2017.

[44] J. Liang, J. Cao, Y. Fan, K. Zhang, R. Ranjan, Y. Li, R. Timofte, and L. Van Gool, “Vrt: A video restoration transformer,” *arXiv preprint arXiv:2201.12288*, 2022.

[45] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 764–773.

[46] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. Price, S. Cohen, and T. Huang, "Youtube-vos: Sequence-to-sequence video object segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018.

[47] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2016.

[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[49] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, 2004.

[50] T. Wang, M. Liu, J. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, "Video-to-video synthesis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018.

[51] W. Lai, J. Huang, O. Wang, E. Shechtman, E. Yumer, and M. Yang, "Learning blind video temporal consistency," in *Proc. Eur. Conf. Comput. Vis.*, 2018.