

Structure-Guided Deep Video Inpainting

Chaoqun Wang, Xuejin Chen, *Member, IEEE*, Shaobo Min, Jiaping Wang, *Member, IEEE*, and Zheng-Jun Zha *Member, IEEE*

Abstract—A fundamental challenge in video inpainting is the difficulty of generating video contents with fine details, while keeping spatio-temporal coherence in the missing region. Recent studies focus on synthesizing temporally smooth pixels by exploiting the flow information, while ignoring maintaining the semantic structural coherence between frames. This makes them suffer from over-smoothing and blurry contours, which significantly reduce the visual quality of inpainting results. To address this issue, we present a novel structure-guided video inpainting approach that enhances temporal structure coherence to improve video inpainting results. In contrast to directly synthesizing the missing pixel colors, we first complete edges in the missing regions to depict scene structures and object shapes via an edge inpainting network with 3D convolutions. Then, we replenish textures using a coarse-to-fine synthesis network with a structure attention module (SAM), under the guidance of the synthesized edges. Specifically, our SAM is designed to model the semantic correlation between video textures and structural edges to generate more realistic content. Besides, motion flows between neighboring frames are employed to enhance temporal consistency for self-supervision during training the edge inpainting and texture inpainting modules. Consequently, the inpainting results using our approach are visually pleasing with fine details and temporal coherence. Experiments on the YouTubeVOS and DAVIS datasets show that our method obtains state-of-the-art performance under diverse video inpainting settings.

Index Terms—Video inpainting, Structure guidance, Flow Assistance.

I. INTRODUCTION

Video inpainting aims to recover the missing content of a corrupted video and assists lots of practical applications, *e.g.*, video restoration and watermarking removal. High-quality video inpainting requires not only realistic structures with visual details but also temporal consistency. Though great progress has been made in 2D image inpainting using deep learning techniques [1], [2], [3], directly applying these approaches to each frame individually for video inpainting will lead to flaws, flickers, and jitters due to the additional time dimension.

Traditional video inpainting methods employ a patch composition framework that composites content in the missing regions via patches by exploiting complementary information across neighboring frames [4], [5], [6]. These methods rely heavily on the hypothesis that the missing content in the corrupted region appears in neighboring frames, which greatly

limits their generalization ability. Recently, deep-learning-based methods have achieved great performance improvement in video inpainting [7], [8], [9], [10], [11], [12]. A straightforward solution is to utilize 3D convolution layers to extract spatio-temporal features and predict missing contents with smooth motion [7]. To obtain temporally smooth results, contextual information from previous frames is aggregated to synthesize corrupted regions of target frame using a recurrent feedback scheme [8], [10], pixel propagation guided by completed flows [9], or copying similar contents from neighbouring frames [11], [12]. By integrating motion guidance, these methods pay more attention to temporal smoothness; however, structure rationality and object details have not been well explored.

Without definite representation and generation of the target image structures, these methods tend to produce over-smoothed regions. Similar observations have been obtained in the image inpainting task [2], [13]. To solve this problem, two-step methods are proposed to complete object contours [2] or edge maps [13] first as auxiliary information to guide texture synthesis later in image inpainting. However, when applying these edge-first image inpainting methods to video inpainting, it brings another challenge in generating temporally coherent structures while human vision is significantly sensitive to temporal discontinuity that frequently occurs at edges.

In order to simultaneously hallucinate detailed image structures and preserve temporal coherence in video inpainting, we present a novel structure-guided video inpainting approach that effectively exploits the spatio-temporal structure information to improve the quality of video inpainting. Compared with previous video inpainting methods that only consider motion guidance, we explore the correlation between structure, texture, and motion to complete the missing region with reasonable structure, rich visual details, and temporal coherence, as shown in Fig. 1. First, we design an edge inpainting network (ENet) to predict sparse edges in the missing region to represent the target structure for each frame by exploiting the spatio-temporal neighboring information from adjacent frames. Then, under the guidance of completed edges, we employ a texture inpainting network (TexNet) to fill the missing region via a coarse-to-fine architecture and a structure attention module (SAM). Specifically, the SAM is designed to guide the texture generation of TexNet by capturing the latent spatial relevance between video textures and the synthesized edges to achieve fewer cracks and more realistic object contours. Furthermore, to enhance the temporal coherence of synthesized frames, we employ motion flows for consistency check of both edge maps and inpainted frames during the training stage. The ground truth optical flow is exploited to guide both ENet and TexNet to generate temporally smooth

C. Wang, X. Chen, S. Min, and Z. Zha are with the National Engineering Laboratory for Brain-inspired Intelligence Technology and Application, University of Science and Technology of China, Hefei, 230026, China.

J. Wang is with the Peng Cheng Laboratory, Shenzhen, 518000, China.

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the author. The material includes ten clips of the inpainted videos described in the paper in standard avi format. The total size of the videos is 40 MB. Contact xjchen99@ustc.edu.cn for further questions about this work.

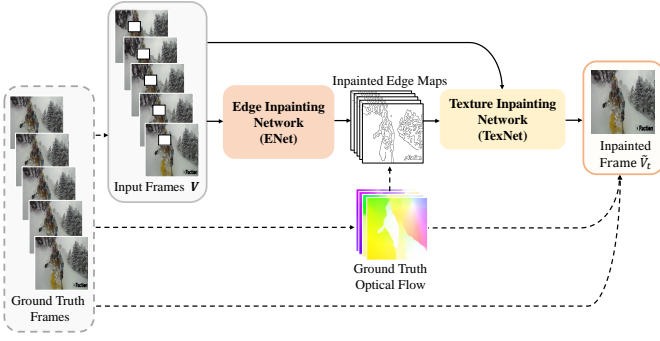


Fig. 1. Overview of our structure-guided video inpainting network. We first complete the edges in the corrupted region by aggregating information from neighboring frames to represent the target structure using the ENet. Then, the TexNet synthesizes missing textures under the structural guidance of inpainted edges. Besides, the ground truth optical flow between frames is utilized during the training stage for both ENet and TexNet to enforce temporal coherence of completed contents, as illustrated by dotted lines.

edge maps and texture results via edge consistency and frame warping losses. Consequently, the inpainted frames using our approach are not only temporally consistent, but also more complete in structure and rich in visual details.

We conduct a series of experiments on the YouTubeVOS and DAVIS datasets under various mask settings. The results show that the proposed method obtains new state-of-the-art inpainting performance both quantitatively and qualitatively. In summary, our technical contributions are three-fold:

- We propose a novel structure-guided video inpainting method which integrates scene structure, texture, and motion to complete the missing region with realistic structure, rich visual details, and temporal coherence.
- A structure attention module is designed to capture the correlation between hallucinated edges and video textures, which can provide better structural guidance for texture synthesis.
- Flow-guided edge and frame consistency constraints are developed to enhance the temporal coherence of both completed edges and video frames.

II. RELATED WORK

Image and video inpainting have been studied for decades. In this section, we mainly discuss four categories of related works, including (a) traditional methods of image inpainting, (b) patch-based video inpainting, (c) CNN-based image inpainting, and (d) deep video inpainting.

a) Traditional Image Inpainting Approaches: Traditional methods of image and video inpainting can be divided into two categories, diffusion-based and patch-based methods. Diffusion-based methods [14], [15], [16] gradually propagate contents from surrounding areas to the missing region. However, this category of methods fails to handle large holes due to its assumption of local smoothness. Patch-based image inpainting methods, also called exemplar-based methods [17], [18], [19], [20], are more widely studied. They formulate the completion task as a patch-based optimization problem. Barnes *et al.* [21] employ approximate nearest neighbor algorithm to

fill the damaged regions. Sangeetha *et al.* [22] propose to propagate both linear structure and two-dimensional texture into the target region. Ružić *et al.* [23] introduce Markov random field to search the most matched candidates. Ding *et al.* [24] employ nonlocal texture similarity and local intensity smoothness to produce natural-looking results. Besides, some patch-based methods utilize low rank approximation. For example, Guo *et al.* [25] propose a simple two-stage low rank approximation to recover the corrupted region, which avoids time-consuming iterations. Lu *et al.* [26] adopt gradient-based low rank approximation. These patch-based methods fill the missing content by borrowing and aggregating the most similar patches based on low-level image features from known regions. However, they usually fail when there is insufficient information in known regions or image textures are too complicated.

b) Patch-Based Video Inpainting: Patch-based methods are also widely studied for video inpainting [27], [28], [29], [30]. They search similar patches and borrow appearances from known regions across neighboring frames to synthesize the unknown content. Wexler *et al.* [27] constrain masked regions to synthesize coherent structures with respect to reference examples based on local structures. Umeda *et al.* [29] propose using directional median filter as complementation of patch-based filling. Newson *et al.* [6] extend the 2D PatchMatch algorithm [21] into 3D version to improve video inpainting quality. Xu *et al.* [30] first complete motion field to guide the patch composition for video background completion. Huang *et al.* [31] jointly estimate optical flow and textures to promote temporal coherence. Another group of methods separate foreground and background apart, and then deal with the two parts with different algorithms, respectively. Ghanbari *et al.* [32] first separate foreground and background in videos, and fills the two parts accordingly with the help of contours. Xia *et al.* [33] make use of Gaussian mixture models to distinguish moving foreground and still background, and process them separately. However, in these patch-based video inpainting methods, the patch-searching process suffers from high computational complexity, thus limiting their usage in practical applications.

c) CNN-based Image Inpainting: Recently, deep learning methods have achieved tremendous progress in the field of computer vision. The tasks of image and video inpainting also have witnessed great promotion thanks to the capability of deep neural networks to capture high-level semantic information in images and videos. A convolution neural network (CNN) is first introduced to directly synthesize image contents in the masked regions for image denoising and inpainting in [34]. To improve the photorealism of the synthesized content, a generative adversarial network is employed [35]. Then, Yang *et al.* [36] take advantages of multi-scale representation to boost details generation. Multiple discriminators are used to constrain both global and local coherence of image contents [37]. Yu *et al.* [38] propose a contextual attention module in capturing long-range information. Subsequent approaches solve more specific problems in image inpainting, for example, inpainting irregular holes with partial convolution [39] or employing gated convolution [1] for dynamic feature selection. While these methods tend to generate over-smoothed and

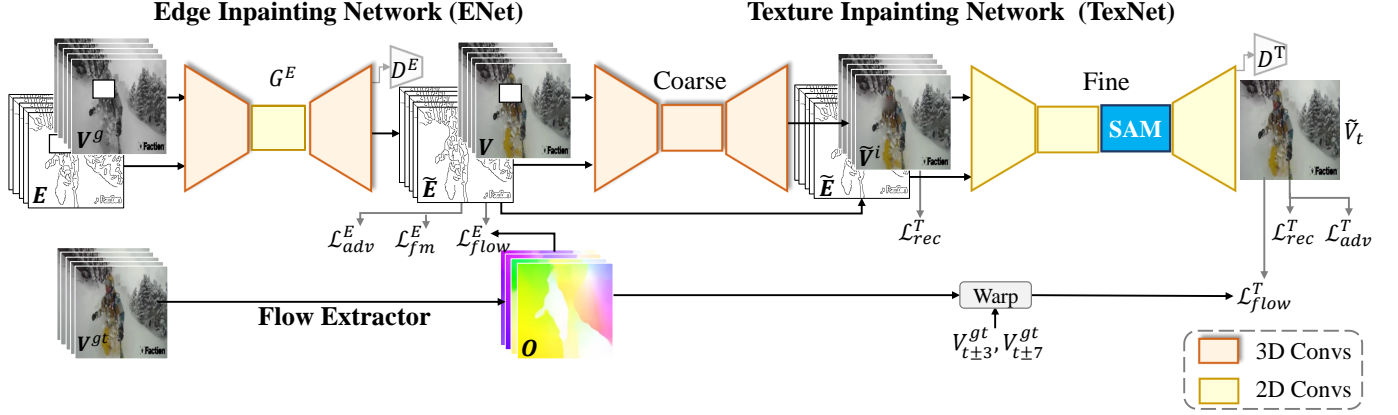


Fig. 2. The detailed architecture of our network. ENet adopts an encoder-decoder architecture to complete edges in the missing regions. TexNet utilizes a coarse-to-fine manner to inpaint the final frames. Ground truth optical flow between adjacent frames is employed in edge consistency loss and frame warping loss to enforce temporal coherence.

blurry results, a two-stage approach is proposed to hallucinate edges first and then fill image colors using the edges as a prior [13]. Similarly, Xiong *et al.* [2] predict contours of foreground objects to guide the inpainting process of masked regions. Though high-quality static images with reasonable structure can be generated using these edge-based methods, simply extending it from image inpainting to video inpainting by 3D convolutions inevitably fails because of no guarantee on the temporal coherence, especially on the high-frequency signals. Besides, these methods simply utilize edges and contours as one additional channel of the image inpainting network, without exploiting the mutual relationship between edges and textures. In comparison, we design a structure attention module to better explore the structural guidance in synthesizing textures.

d) Deep Video Inpainting: Several video inpainting methods based on deep neural networks have been proposed recently, to fully utilize useful complementary information in neighboring frames. The first deep-learning-based video inpainting method is CombCN [7], which jointly learns temporal structure and spatial details via 3D convolutions. To enforce temporal coherence, features from neighboring frames are collected and refined to synthesize the missing content with recurrent feedback [8], [10]. OPN [12] uses the non-local formulation to calculate the similarity between pixels in reference and target frames, and then copies the contents based on the similarity. Similarly, Copy-and-Paste [11] copies matching contexts in aligned reference frames and pastes them to the corrupted region. However, both OPN and Copy-and-Paste only consider the texture correlation between neighboring frames, while ignoring high-level structures. In comparison, we explore the correlation between structures and textures to achieve high-quality inpainting results. Instead of filling pixel colors directly using CNNs, a deep flow-guided inpainting network is proposed in [9] to estimate optical flow first in the missing region and then propagate pixel colors based on the completed flow. However, these existing methods usually suffer from blurs and structural cracks in the synthesized frames since it is non-trivial to maintain fine details and sharp

edges simultaneously when predicting temporally coherent pixel colors. In comparison, we propose to explicitly complete the target structure using edges, which are efficient due to their sparsity. To utilize structural information more effectively, we also introduce a structure attention mechanism. Under the structural guidance, more visually pleasing results could be synthesized with plausible structure and fine details.

III. APPROACH

Our goal is to recover the missing contents in a corrupted video, with consideration of fine details and temporal consistency. To produce a completed frame \tilde{V}_t at time t , we take total T ($T = 5$) frames $V = \{V_{t-7}, V_{t-3}, V_t, V_{t+3}, V_{t+7}\}$, as input to our inpainting network in each data batch. The corrupted regions in the T frames are indicated by binary masks $M = \{M_{t-7}, M_{t-3}, M_t, M_{t+3}, M_{t+7}\}$ where $M_t(p) = 1$ indicates that pixel p is corrupted in frame V_t .

The detailed architecture of our inpainting network is shown in Fig. 2. It consists of three main components. The first part is an edge inpainting network (ENet) for structure inference that recovers edges in the missing regions of the input frames. The second part is a coarse-to-fine texture inpainting network (TexNet) that aims to complete the missing contents with visual details under the guidance of hallucinated edges. The third part leverages the ground truth motion flow as auxiliary constraints to enforce temporal coherence of both the completed edge maps and textures at the training stage.

A. Edge Inpainting Network

The edge inpainting network (ENet) completes image edges in the missing regions to depict scene structures and object shapes. From T neighboring corrupted frames V as well as the corresponding set of binary masks M , a Canny edge detector is first applied to extract the corresponding edge maps E in the uncorrupted regions. The input of ENet is the concatenation of the grayscale version of T corrupted frames V^g , initial edge maps E , and their corresponding masks M . As shown in Fig. 2, our ENet, denoted by G^E , is composed of a 3D

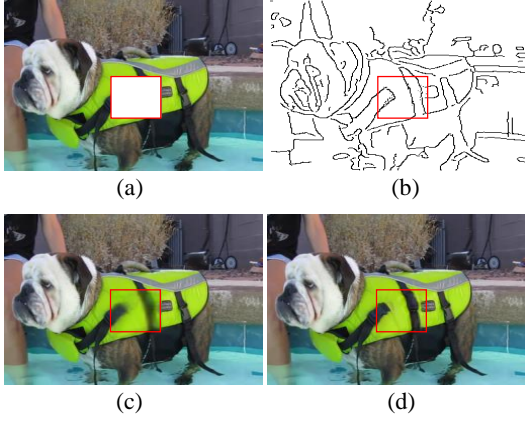


Fig. 3. To inpaint missing regions in a corrupted frame (a), our ENet first completes corresponding sparse edges (b), which well represent the structure of the missing contents. Then TexNet progressively replenishes textures under the guidance of synthesized edges from coarse (c) to fine (d).

encoder, several 2D residual blocks, and a 3D decoder. The 3D encoder and decoder are designed to capture temporal coherence, which takes large computation consumption. The intermediate 2D residual blocks are efficient in aggregating spatial contexts via a large spatial receptive field. Notably, under a limited number of network parameters, the intermediate part of ENet can accommodate more 2D convolutions than 3D convolutions, thereby obtaining larger spatial receptive fields. This hybrid 2D and 3D convolution network achieves a good balance between spatial and temporal coherence. The inpainted T edge maps $\tilde{\mathbf{E}} = \{\tilde{E}_{t-7}, \tilde{E}_{t-3}, \tilde{E}_t, \tilde{E}_{t+3}, \tilde{E}_{t+7}\}$ are obtained by:

$$\tilde{\mathbf{E}} = G^E(\mathbf{E}, \mathbf{V}^g, \mathbf{M}). \quad (1)$$

Without any auxiliary information, the edge inpainting network ENet can be trained adversarially by:

$$\min_{G^E} \max_{D^E} (\mathcal{L}_{adv}^E + \lambda_1 \mathcal{L}_{fm}^E), \quad (2)$$

where the discriminator D^E follows the 70×70 PatchGAN architecture [40]. \mathcal{L}_{adv}^E and \mathcal{L}_{fm}^E are the adversarial loss and feature matching loss respectively. λ_1 is a hyper-parameter to balance the two terms. \mathcal{L}_{adv}^E tends to make the inpainted edge maps follow the distribution of real edge maps by

$$\begin{aligned} \mathcal{L}_{adv}^E = & \mathbb{E}_{(E_t^{gt}, V_t^g)} [\log D^E(E_t^{gt}, V_t^g)] \\ & + \mathbb{E}_{(\tilde{E}_t, V_t^g)} [\log(1 - D^E(\tilde{E}_t, V_t^g))]. \end{aligned} \quad (3)$$

\mathcal{L}_{fm}^E evaluates the feature-level similarity between ground truth and predicted edge maps, which is defined by:

$$\mathcal{L}_{fm}^E = \sum_{t=1}^T \sum_{k=1}^L \frac{1}{N_k} \left\| D_k^E(E_t^{gt}, V_t^g) - D_k^E(\tilde{E}_t, V_t^g) \right\|_1, \quad (4)$$

where D_k^E is the k -th layer in the L -layer D^E , and N_k is the element number of the output of D_k^E . We use the feature similarity loss \mathcal{L}_{fm}^E instead of L1 reconstruction loss because that L1 reconstruction loss is sensitive to small drifts and noises for sparse edge maps. As a result, the proposed hybrid 3D and 2D convolution architecture and the adversarial loss

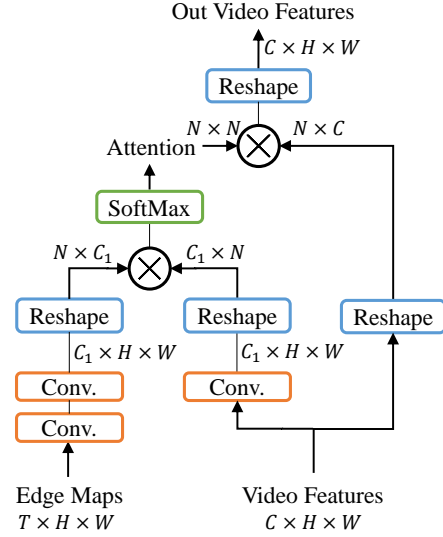


Fig. 4. Architecture of the structure attention module. C is number of channels of the input video features, and $N = H \times W$. \otimes represents matrix multiplication. Usually, we set $C_1 = C/8$.

enable our ENet to hallucinate the missing edges accurately and efficiently. An example of the generated edge map is given in Fig. 3 (a) and (b). The stripe pattern on the dog is well inferred from neighboring frames.

B. Edge-Guided Texture Inpainting Network

With the T completed edge maps $\tilde{\mathbf{E}}$ for the T neighboring frames \mathbf{V} , we then fill the image texture using TexNet. Notably, the image structure, e.g., object shapes, is well represented by the completed edge maps $\tilde{\mathbf{E}}$. Thus, it becomes easier to fill the missing texture with the structural guidance of $\tilde{\mathbf{E}}$. To synthesize realistic frame textures, our TexNet adopts a coarse-to-fine architecture, as shown in Fig. 2. Specifically, TexNet G^T consists of a coarse inpainting network G_c^T and a refinement network G_r^T . First, the coarse inpainting network consists of a set of 3D convolutions to capture the spatio-temporal contexts from the hallucinated edge maps and the corrupted input frames to produce a rough completion $\tilde{\mathbf{V}}^i$ for the T frames with colors and textures. The input of G_c^T is the concatenation of T frames \mathbf{V} , the synthesized edge maps $\tilde{\mathbf{E}}$, and the masks \mathbf{M} :

$$\tilde{\mathbf{V}}^i = G_c^T(\mathbf{V}, \tilde{\mathbf{E}}, \mathbf{M}), \quad (5)$$

where $\tilde{\mathbf{V}}^i$ is the output rough inpainting frames. Then, the refinement network takes the concatenation of the rough inpainting results $\tilde{\mathbf{V}}^i$, the synthesized edge maps $\tilde{\mathbf{E}}$, and the masks \mathbf{M} as inputs to further refine texture details in $\tilde{\mathbf{V}}^i$ with the structural guidance of edges $\tilde{\mathbf{E}}$. Notably, only 2D convolutional layers are used in the refinement network to improve inference efficiency. The refinement network finally reconstructs current frame:

$$\tilde{\mathbf{V}}_t = G_r^T(\tilde{\mathbf{V}}^i, \tilde{\mathbf{E}}, \mathbf{M}). \quad (6)$$

Besides of taking $\tilde{\mathbf{E}}$ as an auxiliary input in the refinement network, we further design a structure attention module (SAM)

to fully encode the structural information. The detailed implementation of SAM is given in Fig. 4. The input of SAM consists of the intermediate video features extracted from the rough inpainting frames \tilde{V}_t and the edge maps \tilde{E} . First, the intermediate video features and embedded edge features are interacted to calculate the latent structure-texture correlation via matrix multiplication. After a SoftMax operation, the normalized attention map is obtained while it conveys the correlation between sparse structures and video features. The normalized attention map is applied to the intermediate video features, and the structure information is thus embedded in TexNet, which can better extract useful structural information from edges. The main difference between our SAM and self attention module [41] is that self attention only considers texture correlation, while our SAM focuses on edge-texture correlation to overcome the over-blurry problem in video inpainting. By introducing structural guidance in the refinement network, the inpainted content by TexNet becomes more realistic, as Fig. 3 (d) shows.

The coarse inpainting network and the refinement network in TexNet are trained end-to-end in an adversarial manner by:

$$\min_{G^T} \max_{D^T} (\mathcal{L}_{rec}^T + \mathcal{L}_{adv}^T), \quad (7)$$

where the discriminator D^T follows the same architecture as D^E . Inspired by [13], the first term \mathcal{L}_{rec}^T is the l_1 -reconstruction loss to measure the difference between predicted video frames and the ground truth video frames V^{gt} . Differently, we penalize both the single refined frame \tilde{V}_t and the coarse predictions of T frames \tilde{V}^i by:

$$\begin{aligned} \mathcal{L}_{rec}^T = & \frac{1}{\|M_t\|_1} \left\| (\tilde{V}_t - V_t^{gt}) \odot M_t \right\|_1, \\ & + \lambda_2 * \frac{1}{\|M\|_1} \left\| (\tilde{V}^i - V^{gt}) \odot M \right\|_1, \end{aligned} \quad (8)$$

where V_t^{gt} denotes the ground truth frame of time t , and M_t is the corresponding binary mask. The extra adversarial loss \mathcal{L}_{adv}^T is introduced in Eq. (7) to promote the visual realism of the generated frame \tilde{V}_t by:

$$\mathcal{L}_{adv}^T = \mathbb{E}[\log D^T(V_t^{gt})] + \mathbb{E}[\log(1 - D^T(\tilde{V}_t))]. \quad (9)$$

The coarse and refinement networks can be effectively trained jointly with the combined loss of \mathcal{L}_{rec}^T and \mathcal{L}_{adv}^T . The results in Fig. 3 (c) and (d) show that the coarse network completes the missing content with rough textures under the structure guidance of inpainted edges and the refinement network exactly refines the inpainting results with more sharp contours and realistic textures.

C. Flow-Guided Temporal Coherence Enhancement

Besides the structural guidance, motion information is also considered in our method to enhance temporal consistency among the recovered frames. We employ optical flows in the training stage of both ENet and TexNet using self-supervision of neighboring frames. A set of flow maps O between the current frame V_t^{gt} and its neighboring frames are generated using a pre-trained flow extraction network, such as

FlowNet2.0 [42]. Specifically, O consists of four flow maps ($O_{t \Rightarrow t-7}, O_{t \Rightarrow t-3}, O_{t \Rightarrow t+3}, O_{t \Rightarrow t+7}$).

In terms of the ENet which completes the missing edges, O is used to first warp the neighboring edge maps to the current frame, and then compute the consistency between neighboring edge maps. A flow-guided edge consistency loss is defined as:

$$\mathcal{L}_{flow}^E = \sum_k \frac{1}{\|M_t\|_1} \left\| (\tilde{E}_t - \phi(O_{t \Rightarrow t+k}, E_{t+k}^{gt})) \odot M_t \right\|_1, \quad (10)$$

where $\phi(O_{t \Rightarrow t+k}, E_{t+k}^{gt})$ is the warping operation which warps a neighboring edge map E_{t+k}^{gt} to the target frame according to the estimated optical flow $O_{t \Rightarrow t+k}$. k denotes the index of neighboring frames ($k \in \{-7, -3, +3, +7\}$). By introducing this additional flow-guided edge consistency loss \mathcal{L}_{flow}^E when training ENet, the original loss function (Eq. (2)) becomes:

$$\min_{G^E} \max_{D^E} (\mathcal{L}_{adv}^E + \lambda_1 * \mathcal{L}_{fm}^E + \mathcal{L}_{flow}^E). \quad (11)$$

Similarly, when we train the TexNet, we further enforce the temporal coherence of synthesized neighboring frames via a flow-guided texture consistency \mathcal{L}_{flow}^T by:

$$\mathcal{L}_{flow}^T = \sum_k \frac{1}{\|M_t\|_1} \left\| (\tilde{V}_t - \phi(O_{t \Rightarrow t+k}, V_{t+k}^{gt})) \odot M_t \right\|_1, \quad (12)$$

where $k \in \{-7, -3, 3, 7\}$, and $\phi(O_{t \Rightarrow t+k}, V_{t+k}^{gt})$ warps the neighboring frame V_{t+k}^{gt} to the target frame using the estimated flow $O_{t \Rightarrow t+k}$. The flow warping loss \mathcal{L}_{flow}^T is added to the objective function of TexNet, which converts Eq. (7) to:

$$\min_{G^T} \max_{D^T} (\mathcal{L}_{rec}^T + \mathcal{L}_{adv}^T + \mathcal{L}_{flow}^T). \quad (13)$$

After adding the motion guidance to both the training processes of ENet and TexNet, the temporal consistency can be enhanced in both edge generation and texture inpainting. Since the motion is only used in the training stage, the inference of the proposed structure-guided inpainting method is efficient and effective.

IV. EXPERIMENTS

To evaluate the effectiveness of our approach, we conduct a series of comparison experiments and ablation studies on two widely used datasets, *i.e.*, YouTubeVOS [43] and DAVIS [44], under different settings.

A. Experimental Settings

Mask Setting. Considering different real-world applications, we test four kinds of mask settings in this paper, which are different in shapes and positions of the missing regions.

- 1) Fixed square mask: The size and position of the missing square regions are fixed through the whole video.
- 2) Moving square mask: The position and size of the square masks change over frames.
- 3) Free-form mask: We apply irregular masks that imitate hand-drawn masks on each frame, following [39].
- 4) Foreground object mask: This type of mask is defined to line out foreground objects in videos and used for testing object removal.

TABLE I

COMPARISONS WITH SIX STATE-OF-THE-ART METHODS ON YOUTUBEVOS. OUR METHOD OUTPERFORMS ALL OTHER METHODS ON THREE VIDEO QUALITY METRICS, WITH COMPARABLE TEMPORAL SMOOTHNESS AND HIGHER INFERENCE SPEED.

	Fixed Square Mask				Moving Square Mask				Free-Form Mask				Inference Speed (fps)
	PSNR	SSIM	FID	Tem-ERR	PSNR	SSIM	FID	Tem-ERR	PSNR	SSIM	FID	Tem-ERR	
Edge-Connect [13]	28.6446	0.9484	38.2116	0.02403	30.7478	0.9647	16.2739	0.02807	25.6693	0.9088	43.0366	0.04130	22.81
CombCN [7]	27.9668	0.9515	40.7199	0.02252	31.5776	0.9678	13.8383	0.02443	32.1862	0.9626	19.1191	0.03241	8.1634
DVI [8]	28.0846	0.9468	39.9377	0.01817	36.8598	0.9728	7.2315	0.01946	33.5549	0.9646	9.3797	0.02188	1.2275
DFVI [9]	29.0531	0.9497	32.8860	0.02025	37.8241	0.9772	6.3746	0.02258	32.6287	0.9618	11.1501	0.02358	0.5620
Copy-and-Paste [11]	27.8722	0.9040	35.5054	0.02282	33.0092	0.9388	10.5099	0.02500	30.2559	0.9106	31.3721	0.02968	1.4741
OPN [12]	29.2663	0.9149	37.9550	0.02328	33.3254	0.9381	12.1887	0.02498	31.7570	0.9211	24.0585	0.02782	3.8239
Ours	30.0590	0.9543	27.2431	0.02082	38.8186	0.9824	2.3455	0.02388	35.9613	0.9721	5.8694	0.02278	5.1546

TABLE II
ENET ARCHITECTURE DETAILS.

Network	Arch	Operation	Kernel size	Output channels
G^E	Encoder	Conv3D	(3,7,7)	[64]
		Conv3D	(3,4,4)	[128,256]
	ResnetBlock	DilatedConv2D	(3,3)	[256]*8
	Decoder	ConvTranspose3D	(3,4,4)	[128,64]
		ConvTranspose3D	(3,7,7)	[1]
D^E		Conv2D	(4,4)	[64,128,256,512,1]

TABLE III
TEXNET ARCHITECTURE DETAILS. WE USE GATED CONVOLUTION FOR THE WHOLE NETWORK, WHICH IS OMITTED FOR SPACE SAVING. DECONVBLOCK DENOTES DECONV+CONV.

Network	Arch	Operation	Kernel size	Output channels
G^T	Coarse	Conv3D	(3,5,5)	[16]
		Conv3D	(3,3,3)	[32,64,128]
		DilatedConv3D	(3,3,3)	[256,256,256]
		Conv3D	(3,3,3)	[128]
		DeConvBlock3D	(3,3,3)	[32,3]
	Refine	Conv2D	(5,5)	[64]
		Conv2D	(3,3)	[128,128,256,256]
		DilatedConv2D	(3,3)	[256,256,256,256]
		Conv2D	(3,3)	[256]
		SAM	-	[256]
		Conv2D	(3,3)	[256]
		DeConvBlock2D	(3,3)	[128,32]
		Conv2D	(3,3)	[3]
D^T		Conv2D	(4,4)	[64,128,256,512,1]

Dataset. YouTubeVOS and DAVIS are widely used for evaluating video inpainting performance in recent studies. YouTubeVOS consists of 4,453 video clips that contain more than 70 categories of common objects. The videos are split into three parts, 3,471 for training, 474 for validation, and 508 for testing. Since YoutubeVOS has no dense foreground mask annotations, we only use it for evaluation under mask settings (1), (2), and (3). DAVIS dataset contains 90 video sequences that are annotated with foreground object masks for testing and 60 unlabeled videos for training.

Implementation details. All the models are tested on a TITAN X (Pascal) GPU with frame size 256×256 . For training data, we randomly sample a training clip every 40 frames from each video in the dataset. Our training process consists of three steps. First, we train ENet with learning rate set as $1e-4$ for G^E and $1e-5$ for D^E . Then we train TexNet while fixing ENet. The learning rate is set as $1e-4$ for G^T , and $4e-4$ for D^T . We first train ENet and TexNet without flow-constrained losses, and then add the flow losses to finetune the networks. An Adam optimizer with $\beta = (0.9, 0.999)$ is used

for all sub-network training. As for the hyper-parameters in the training losses, we set $\lambda_1 = 10.0$ in Eq. (11), $\lambda_2 = 0.2$ in Eq. (8), which will be analyzed later. The detailed network parameters of ENet and TexNet are listed in the Table II and Table III, respectively. During inference, we sequentially feed five neighboring frames with gaps into the inpainting network and output the reconstructed frame \tilde{V}_t in one forward pass.

Evaluation Metrics. Different data preparations and evaluation metrics are used according to mask settings. We randomly generate masks for training videos in terms of mask settings (1), (2), and (3). Four commonly-used metrics, including structural similarity index (SSIM) [45], peak signal-to-noise ratio (PSNR), Fréchet Inception Distance (FID) [46], and temporal warping error (Tem-ERR) [10] are used to quantitatively evaluate the performance of our method. Specifically, PSNR, SSIM, and FID focus on frame generation quality, while Tem-ERR is used to assess the temporal smoothness. For each training video at mask setting (4), we randomly select a test video from the 90 test videos in the DAVIS dataset and apply its masks for the current training video with random rotation, scaling, and translation. Our inpainting network is first trained on the YoutubeVOS dataset and then finetuned on the DAVIS dataset. Since there is no ground truth video for mask setting (4) of foreground object removal, quantitative evaluation is infeasible. Instead, a user study is conducted.

B. Evaluation of Video Inpainting on YouTubeVOS

We compare the proposed method with six state-of-the-art video inpainting methods [13], [7], [8], [9], [11], [12] for the first three mask settings on the YouTubeVOS dataset. We train [13], [9] using their published codes and re-implement [7] according to their paper. As for [8], [11], [12], we use the officially provided model.

The quantitative results and inference speeds are reported in Table I. It can be seen that our method outperforms state-of-the-art methods on most of the metrics of generation quality. First, in terms of three metrics of PSNR, SSIM, and FID about inpainting quality, our method outperforms all other methods, demonstrating the effectiveness of introducing structure guidance into video inpainting. Especially, compared with the 2D image inpainting method Edge-Connect [13], which also predicts edges to represent the target structure, our method greatly increases the completion performance by leveraging neighboring frames to complete edges and synthesize textures. Compared with other flow-based video inpainting methods without edge enhancement, our method

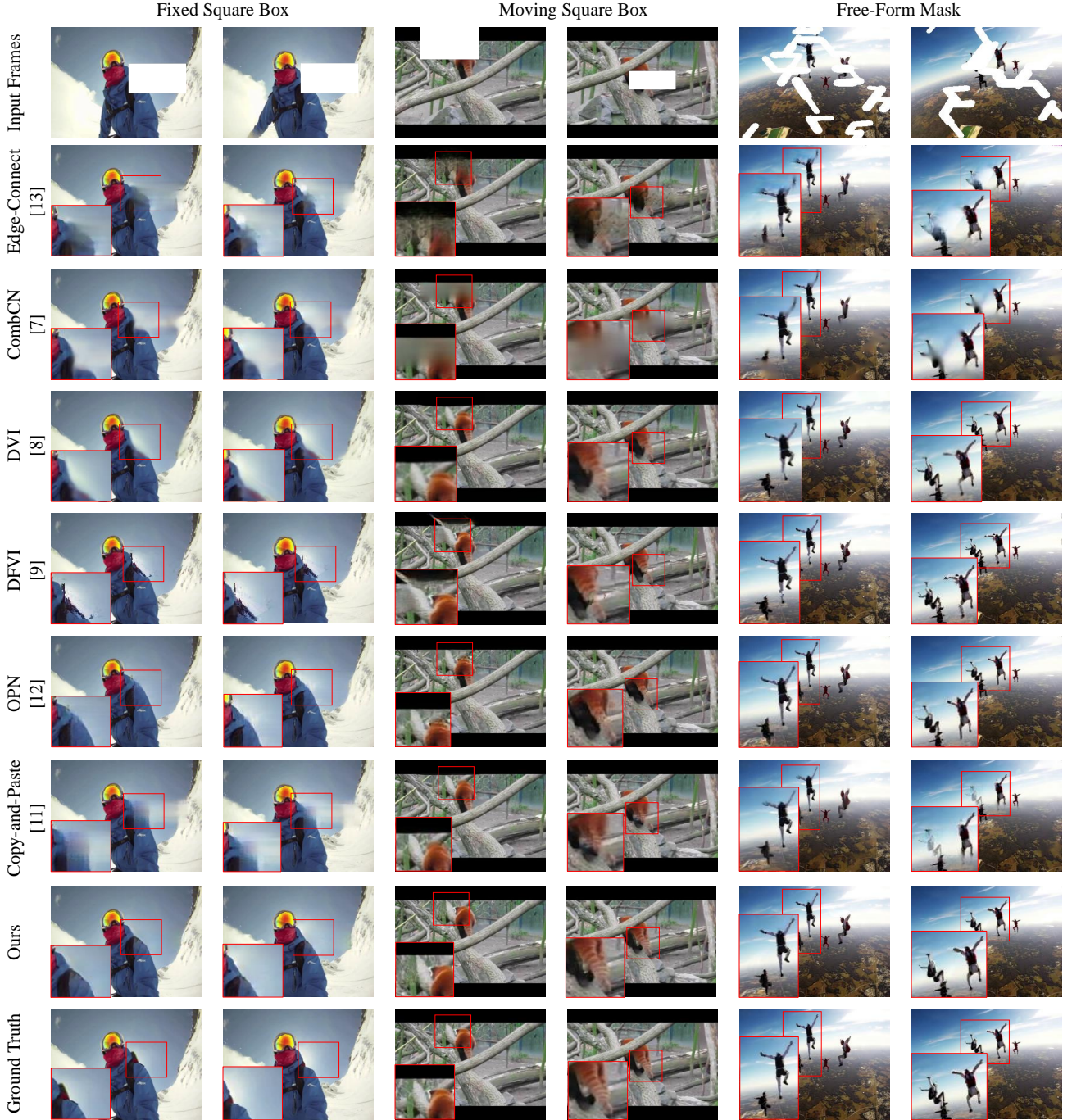


Fig. 5. Comparison with state-of-the-art methods on the YouTubeVOS dataset under different mask settings. Our method produces results with more complete object structures and finer details.

shows great superiority in generating temporally coherent and realistic content.

Second, in terms of Tem-ERR about temporal smoothness, our method outperforms most of the existing methods, proving that the temporal consistency constraint during training can exactly facilitate temporal coherence. Specifically, as shown in Table I, our method is comparable with DFVI [9] and worse than DVI [8]. The performance superiority of DVI mainly comes from the recurrent feedback loop and a memory layer (LSTM) for temporal consistency of inpainted frames.

DFVI iteratively propagates pixel colors according to optical flows. Both DFVI and DVI focuses on designing temporal modules to obtain smooth inpainting results at the inference stage. In comparison, we enforce the temporal coherence of our ENet and TexNet by introducing motion constraints only in training. Therefore, our method is much efficient, e.g., four times faster than DVI [8] and nine times faster than DFVI [9]. The inference speed of the image inpainting method Edge-Connect [13] is faster than our method since it does not consider axillary temporal information between frames.

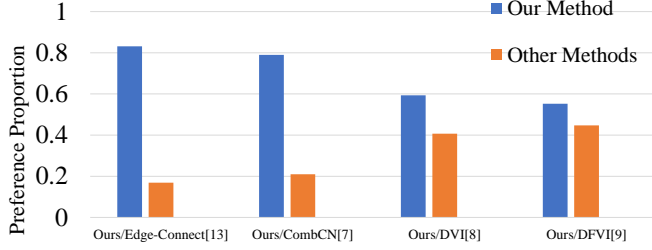


Fig. 6. Results of user study. The restored videos using our method are preferred by more participants compared with other state-of-the-art methods.

Besides, OPN [12] and Copy-and-Paste [11] are two recent methods that leverage non-local correlation between reference and target frames. Compared to the texture interaction between video frames considered in their methods, our SAM models the correlation between sparse edges and realistic textures, leading to better inpainting results and higher inference speed. It shows that the structure guidance introduced edges is critical in video inpainting, and our method is capable of exploiting the correlation between video contents and object structures.

Finally, some inpainting examples are shown in Fig. 5. Compared with existing methods, the inpainting results predicted by our method are more realistic with finer details. The frames completed using our method contain sharper object boundaries. This is achieved by the effectiveness of structure information in video inpainting. Moreover, our method produces temporally smooth results when observing neighboring frames. More comparisons can be found in the supplementary video. In summary, Both the quantitative and qualitative results demonstrate that our method is not a naive extension to utilize structure information in video inpainting. The well-designed network architecture fully exploits structural guidance in video inpainting and brings large performance gain.

C. Object Removal on DAVIS

In regard to the foreground object mask setting that aims to remove undesired objects in videos, there is no ground truth for quantitative evaluation. Therefore, we conduct a user study on the DAVIS dataset to evaluate the visual quality of completion results using our method, compared with the four methods [13], [7], [8], [9]. In each test, we show three videos to the subject at the same time. The original video with red masks indicating objects to remove is shown in the middle, while the inpainting results of our method and one of the other four methods are shown on the two sides in random order. The subjects can watch the videos repeatedly to better evaluate the differences. For each video triplet, the subject is asked to choose which inpainting video is preferred. 44 subjects participated in our user study. Each participant watched averagely 20 triplets. Therefore, each pair of methods is compared about 220 times.

The preference results in the user study are shown in Fig. 6. Comparing to Edge-Connect [13], CombCN [7], DVI [8], our results are preferred by a significantly larger portion of subjects. When comparing with the flow-guided method



Fig. 7. Results of object foreground removal. The red masks in input frames indicate the objects to be removed. Our method produces results with plausible structures and details.

DFVI [9], our method is preferred by 55.24% of the tests. Notably, our method is much faster than DFVI. Fig. 7 shows two examples of object removal using different methods. We can see that the inpainted results generated by our method are visually better than existing methods. Compared to the blurry contents in the results of Edge-Connect, CombCN, DVI, and Copy-and-Paste, our method produces sharp object boundaries and fine visual details. Notably, though the completed contents using DFVI and OPN have sharp edges, the global structure of the human bodies is corrupted. In comparison, our method achieves more intact and plausible structure with fine details. The results demonstrate the importance of utilizing structure information in video inpainting.

D. Ablation Study

To demonstrate the effectiveness of each component in our network, we conduct a series of ablation studies on the YouTubeVOS dataset with the first three mask settings. We test four variants of our model. The baseline model ‘TexNet’ only consists of the coarse-to-fine texture inpainting network without using edge maps as input and no SAM in the refinement module. This model simply integrates neighboring frames to predict the missing content for the current frame. Then we add the edge inpainting network ENet and feed the texture inpainting network with the completed edge maps to get the second model ‘+Edge’. The third model ‘+SAM’ is constructed by adding the structure attention module on the second model. Finally, we add the flow constrained loss during training to get our full model ‘+Flow’. Especially, we only

TABLE IV
ABLATION STUDIES ON YOUTUBEVOS. STRUCTURE INFERENCE, STRUCTURE ATTENTION MODEL, AND FLOW CONSTRAINED LOSS ARE DEMONSTRATED EFFECTIVE IN VIDEO INPAINTING.

	Fixed Square Mask				Moving Square Mask				Free-Form Mask				#Params (M)	GFLOPs	Inference Speed (fps)
	PSNR	SSIM	FID	TEM-ERR	PSNR	SSIM	FID	TEM-ERR	PSNR	SSIM	FID	TEM-ERR			
TexNet	28.0174	0.9494	42.7164	0.02301	33.8131	0.9705	8.2390	0.02713	30.0680	0.9390	20.6358	0.02814	28.85	476.5	7.6335
+Edge	29.5242	0.9520	36.2097	0.02412	37.6630	0.9798	3.5161	0.02579	33.8206	0.9659	6.6651	0.02532	42.26	849.7	5.2356
+SAM	29.9918	0.9533	27.4198	0.02301	38.2433	0.9807	2.5083	0.02535	35.7783	0.9712	5.8786	0.02517	42.33	859.7	5.1546
+Flow	30.0590	0.9543	27.2431	0.02082	38.8186	0.9824	2.3455	0.02388	35.9613	0.9721	5.8694	0.02278	42.33	859.7	5.1546

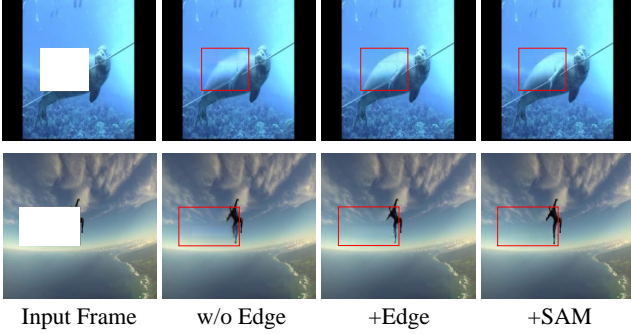


Fig. 8. Effects of structural guidance. By hallucinating edge maps first and then filling texture, we generate more completed and plausible target structure. Sharper boundaries can be obtained using the structure attention module.

add the flow constraints in the training stage, which means it brings no computation costs to the inference process. The quantitative results are reported as in Table IV.

1) *Effect of Structure Clues:* In Table IV, The model ‘+Edge’ brings large improvement over the baseline model. It indicates that sparse edges can provide effective structural guidance in video inpainting. On the other hand, the structure inference module ENet brings extra computational cost compared to the baseline TexNet from 7.6335 *fps* to 5.2356 *fps*. When we further add SAM, extra performance improvement is obtained, demonstrating that the spatial correlation between edges and textures can be better embedded and absorbed by the texture inpainting network than simply feeding the hallucinated edge maps as extra channels into TexNet. Besides, the SAM does not bring too much computational cost, as shown by #Params and GFLOPs in Table IV. This is because SAM takes down-sampled edge maps and video feature maps in low resolution as input. The comparable inference speeds also demonstrate that in the current setting, SAM brings performance gains with affordable cost and can be used in practice. However, when coming to high-resolution inpainting task, SAM may put a higher demand on the computational resource.

Fig. 8 shows the results generated using the three variants. It is obvious that after introducing structural guidance, the inpainted frames become more visually pleasing with sharper object boundaries. Besides, the edges predicted by our method are reasonable and clear, which well represent the image structure and show the strong edge inpainting ability of ENet. Thus, structure clues are effective guidance in video inpainting and facilitate the prediction of more plausible frames with completed and detailed structure.

TABLE V
COMPARISON OF PROPOSED STRUCTURE ATTENTION MODULE (SAM) AND TWO VARIANTS OF ATTENTION MODULES ON YOUTUBEVOS.

	Free-Form Mask		
	PSNR	SSIM	FID
+Edge	33.8206	0.9659	6.6651
+SimATT	34.4321	0.9685	6.3125
+SelfATT	34.6301	0.9655	5.9124
+SAM	35.7783	0.9712	5.8786

2) *Analysis of the Structure Attention Module:* We propose a novel structure attention module (SAM) in TexNet to encode the structure information of edge maps into the texture inpainting. Here, to prove the effectiveness of SAM in exploiting the structure information, we conducted a comparison experiment between the proposed SAM and two other commonly used variants of attention module. SimATT is a simple two-layer attention [47], *i.e.*, using a two-layer convolution network to extract a spatial attention map from predicted edge maps, which is then applied to the same video feature as SAM. SelfATT is a self attention module [41] which takes only video features as input, without edges. As shown in Table V, SimATT performs worse than the proposed SAM since it lacks edge-texture interaction during attention generation. SelfATT produces better results than adding simple attention, but worse than our SAM. By adding structural-texture mutual interaction in our SAM, we obtain better results than SimATT and SelfATT. These results prove that structure information is important in video inpainting, and our SAM is more effective in revealing the potential correlation between structure information in edge maps and video contents.

3) *Visualization of Structure Attention:* To further reveal the effect of our proposed structure attention module (SAM) in the TexNet, we visualize the attention maps produced during the inpainting process for a group of examples in Fig. 9. For each example, we pick a foreground pixel (red) and a background pixel (green) in the missing region, and show their corresponding attention maps. While the coarse inpainting network in TexNet effectively produces most low-frequency signals in the missing regions, the refinement network aims to add high-frequency signals such as visual details and sharp edges for both foreground and background pixels. As shown in Fig. 9 (c) and (d), most of these high-frequency information comes from the edge pixels. Especially, in terms of the foreground pixels, the attention maps show that the refinement network aggregates information mainly from the edge pixels to preserve a consistent and clear object shape, which makes the results visually reasonable, as Fig. 9 (c) shows. In comparison, to complete the background pixels, the

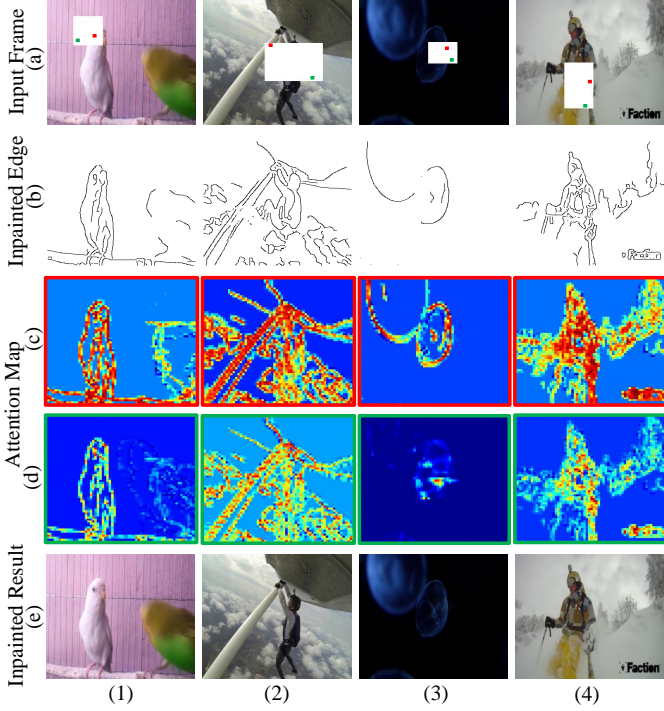


Fig. 9. Visualization of attention maps in the proposed SAM. (a) A foreground pixel (red) and a background pixel (green) in a corrupted frame. (b) The completed edge maps by ENet. (c) and (d) The attention maps show the correlations between the selected foreground/background pixels and other regions respectively. (e) The final inpainting results from the refinement network with structural guidance.

SAM adaptively collects information from different regions with relatively lower correlation with the edge pixels, as shown in Fig. 9 (d). In the first and third examples, SAM derives small weights for those edge regions to complete the background pixels with smooth textures. For the background pixels in the second and fourth examples, high-frequency information is necessary. SAM derives higher weights for edge pixels in the attention maps. The visualization shows that the proposed SAM exploits information from the edge pixels for the refinement network, which is effective in embedding the structural guidance into high-quality texture generation.

4) *Different Architectures of ENet*: We adopt a hybrid 3D+2D convolution architecture for ENet. To justify such a design, we compare our hybrid 3D and 2D architecture with two variants. The first Variant-1 directly replaces the 2D convolutions of ENet with 3D convolutions, which has the same number of layers with our hybrid architecture. The second Variant-2 replaces the 2D convolution layers with fewer 3D convolution layers while keeping similar number of network parameters with our current architecture. The comparison is shown in Table VI. It can be seen that Variant-1 has slightly better performance because full 3D convolutions can capture finer temporal details and introduce spatial information from features of neighbor frames. However, it has about 2.5 times parameters of our hybrid 3D+2D architecture. Variant-2 results in lower generation quality because of its shallower intermediate layers. In comparison, our hybrid 3D+2D convolutions architecture achieves comparable results with Variant-1

TABLE VI
COMPARISON OF DIFFERENT ARCHITECTURES OF ENET.

	Free-Form Mask			#Params (M)
	PSNR	SSIM	FID	
Variant-1	35.8167	0.9714	3.8173	32.29
Variant-2	34.9607	0.9667	5.9657	14.59
Ours	35.7783	0.9712	5.8786	13.41

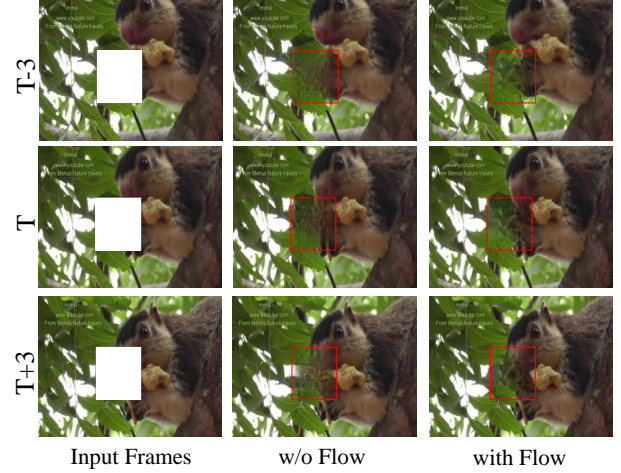


Fig. 10. Inpainting results of three neighboring frames. With the flow constraints during network training, the inpainting results are more temporally consistent without introducing image blurs.

with much fewer parameters, demonstrating its better balance between computational cost and inpainting quality.

5) *Effect of Flow for Temporal Coherence*: We utilize temporal information to smoothen artificial flickers via two developed flow-guided warping losses during training. Table IV shows that quantitative performance is improved on all three mask settings by adding the flow guidance. According to the metric of Tem-ERR, the inpainting results are temporally smoother when adding edge and SAM. Furthermore, adding the flow-based constraints brings additional improvement in temporal consistency. Especially, we only add flow guidance in the training phase. Thus it brings performance gains without extra computation costs during testing. As Fig. 10 shows, the synthesized contents in neighboring frames become more temporally consistent by adding the flow constrained losses. This proves that the proposed two flow-guided constraints in edge and texture inpainting networks are effective in enhancing temporal consistency.

6) *Effects of Different Hyper Parameters*: We conduct experiments to determine the hyper-parameters of λ_1 in Eq. (11) and λ_2 in Eq. (8). We use the integration model of ENet and TexNet in this experiment without SAM and flows. We first train ENet with different values of λ_1 , and then train TexNet with recovered edge maps while fixing ENet. The value of λ_2 is set as 0.2 when testing different λ_1 . When determining λ_2 for TexNet, we set $\lambda_1 = 10.0$.

λ_1 is used in Eq. (11) as a weight of feature matching loss when training ENet. Fig. 11 (left) shows that when increasing λ_1 from 0.0 to 2.0, the performance gain is obvious, which proves that the feature matching loss is effective in generating

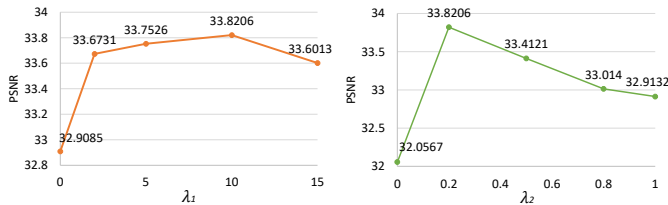


Fig. 11. Comparison of hyper parameters λ_1 and λ_2 . The best result is obtained when $\lambda_1 = 10.0$ and $\lambda_2 = 0.2$.

TABLE VII
COMPARISON OF DIFFERENT INPUT FRAME GAPS.

Frame Gap	Free-Form Mask			
	PSNR	SSIM	FID	Tem-ERR
$\{V_{t-4}, V_{t-2}, V_t, V_{t+2}, V_{t+4}\}$	35.4981	0.9675	6.2889	0.02240
$\{V_{t-15}, V_{t-8}, V_t, V_{t+8}, V_{t+15}\}$	32.2267	0.9505	10.2866	0.02826
$\{V_{t-7}, V_{t-3}, V_t, V_{t+3}, V_{t+7}\}$	35.9613	0.9721	5.8694	0.02278

high-quality edge maps used for the final inpainted results. Then when λ_1 is increased from 2.0 to 10.0, the improvement is slight. The model obtains the best performance at $\lambda_1 = 10.0$. λ_2 in Eq. (8) is the weight of L1 reconstruction loss of the coarse prediction in TexNet. When λ_2 is 0.0, the TexNet is trained without the supervision of the coarse prediction, and the inpainting quality is significantly harmed. It demonstrates that the coarse-to-fine architecture is effective in TexNet. The performance drops when $\lambda_2 > 0.2$, which reflects that the supervision on the fine prediction networks is more important in synthesizing visual details. Therefore, we set λ_2 as 0.2 in all the other experiments.

7) *Different Frame Gaps*: We conducted experiments to evaluate the effects of different frame gaps. In our default setting, we use T frames $\mathbf{V} = \{V_{t-7}, V_{t-3}, V_t, V_{t+3}, V_{t+7}\}$ as input to extract information neighboring frames. In this experiment, we try our inpainting network with smaller gaps, $\mathbf{V} = \{V_{t-4}, V_{t-2}, V_t, V_{t+2}, V_{t+4}\}$, and larger gaps, $\mathbf{V} = \{V_{t-15}, V_{t-8}, V_t, V_{t+8}, V_{t+15}\}$. As reported in Table VII, the smaller gap makes inpainting results temporally smoother (higher Tem-ERR) while the larger gap harms both spatial and temporal coherence, because it may introduce noises from distant frames. We finally choose a trade-off setting with gap of $\{V_{t-7}, V_{t-3}, V_t, V_{t+3}, V_{t+7}\}$ that is able to aggregate sufficient information from neighboring frames and produce temporally smooth results.

8) *Generalization to Different Hole Size*: To investigate the robustness of our method on different hole sizes, we test three hole sizes with fixed square mask setting, including large square in 120×120 , medium square in 60×60 , and small square in 30×30 . The results using different methods are listed in Table VIII. The performances of all the methods drop greatly on large holes. Our method achieves the best performance under most circumstances. The PSNR under large holes of our method is slightly worse than that of OPN [12], which designs an asymmetric attention block to support unlimited spatial-temporal window. However, our method is much better than OPN under other settings. The results show that the performance of our method is stable on different hole sizes.

TABLE VIII
COMPARISONS ON DIFFERENT SIZES OF HOLES.

Hole Size	Fixed Square Mask					
	120×120		60×60		30×30	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Edge-Connect [13]	22.6476	0.8805	30.8471	0.9725	36.0358	0.9908
CombCN [7]	22.0341	0.8710	29.9125	0.9712	34.1423	0.9935
DVI [8]	22.1226	0.8693	28.8842	0.9565	34.5024	0.9780
DFVI [9]	21.9887	0.8750	30.2409	0.9686	37.6599	0.9920
Copy-and-Paste [11]	22.6380	0.8484	29.7965	0.9286	33.9894	0.9474
OPN [12]	23.5713	0.8619	31.2211	0.9338	36.9422	0.9694
Ours	<u>23.2086</u>	0.8876	32.5849	0.9763	42.3711	0.9952

TABLE IX
COMPARISONS ON HIGH RESOLUTION INPAINTING ON SIZE 1024×512 .

	Free-Form Mask		
	PSNR	SSIM	FID
Edge-Connect [13]	29.2517	0.9373	24.5495
CombCN [7]	31.5213	0.9531	15.9324
DVI [8]	32.3845	0.9618	11.7138
DFVI [9]	33.8953	0.9601	12.5124
Copy-and-Paste [11]	31.8301	0.9412	10.7987
Ours	36.7582	0.9790	3.7577

9) *Generalization to Higher Resolution Inpainting*: We also conducted experiments on video inpainting for higher resolutions. Based on the fully convolutional network architecture, we use the model that is trained on low resolution (256×256) to complete videos in higher resolution (1024×512). Results using different methods are reported in Table IX. Notably, OPN [12] is not compared here because of its significantly high memory cost that cannot be deployed on our GPU cluster. Compared to other methods, our method achieves the best performance with great superiority for high-resolution inpainting, demonstrating the effectiveness and robustness of our method.

V. CONCLUSION

In this paper, we propose a novel structure-guided video inpainting approach, which effectively utilizes structural information to recover fine-detailed content in corrupted videos. We first infer the target structure by predicting sparse edges in the missing region using an edge inpainting network. Then under the guidance of hallucinated edges, the missing content can be synthesized using the proposed coarse-to-fine texture network. We propose a structure attention module that effectively exploits the correlation between structures and textures to improve the visual quality by producing more complete structure and visual details. Moreover, the temporal coherence of completed frames is further enhanced by our flow-assisted losses. Experiments on YouTubeVOS and DAVIS datasets under various mask settings demonstrate the effectiveness of our method on video inpainting and restoration tasks.

REFERENCES

- [1] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," *arXiv preprint arXiv:1806.03589*, 2018.
- [2] W. Xiong, J. Yu, Z. Lin, J. Yang, X. Lu, C. Barnes, and J. Luo, "Foreground-aware image inpainting," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

- [3] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [4] K. A. Patwardhan, G. Sapiro, and M. Bertalmio, "Video inpainting under constrained camera motion," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 545–553, 2007.
- [5] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 463–476, 2007.
- [6] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez, "Video inpainting of complex scenes," *SIAM Journal on Imaging Sciences*, vol. 7, no. 4, pp. 1993–2019, 2014.
- [7] C. Wang, H. Huang, X. Han, and J. Wang, "Video inpainting by jointly learning temporal structure and spatial details," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.
- [8] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, "Deep video inpainting," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- [9] R. Xu, X. Li, B. Zhou, and C. C. Loy, "Deep flow-guided video inpainting," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- [10] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, "Deep blind video decactioning by temporal aggregation and recurrence," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2019.
- [11] S. Lee, S. W. Oh, D. Won, and S. J. Kim, "Copy-and-paste networks for deep video inpainting," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4413–4421.
- [12] S. W. Oh, S. Lee, J.-Y. Lee, and S. J. Kim, "Onion-peel networks for deep video completion," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4403–4412.
- [13] K. Nazari, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi, "Edgeconnect: Structure guided image inpainting using edge prediction," in *The IEEE International Conference on Computer Vision Workshops*, 2019.
- [14] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000.
- [15] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," *IEEE Transactions on Image Processing*, vol. 10, no. 8, pp. 1200–1211, 2001.
- [16] G. Sridevi and S. S. Kumar, "Image inpainting based on fractional-order nonlinear diffusion for image reconstruction," *Circuits, Systems, and Signal Processing*, vol. 38, no. 8, pp. 3802–3817, 2019.
- [17] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 882–889, 2003.
- [18] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001.
- [19] X. Li and Y. Zheng, "Patch-based video processing: A variational bayesian approach," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 1, pp. 27–40, 2008.
- [20] J. Zhang, D. Zhao, R. Xiong, S. Ma, and W. Gao, "Image restoration using joint statistical modeling in a space-transform domain," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 6, pp. 915–928, 2014.
- [21] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patch-Match: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics (ToG)*, vol. 28, no. 3, p. 24, 2009.
- [22] K. Sangeetha, P. Sengottuvelan, and E. Balamurugan, "Combined structure and texture image inpainting algorithm for natural scene image completion," *Journal of Information Engineering and Applications*, vol. 1, no. 1, pp. 7–12, 2011.
- [23] T. Ružić and A. Pižurica, "Context-aware patch-based image inpainting using markov random field modeling," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 444–456, 2014.
- [24] D. Ding, S. Ram, and J. J. Rodríguez, "Image inpainting using nonlocal texture matching and nonlinear filtering," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1705–1719, 2019.
- [25] Q. Guo, S. Gao, X. Zhang, Y. Yin, and C. Zhang, "Patch-based image inpainting via two-stage low rank approximation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 6, pp. 2023–2036, 2018.
- [26] H. Lu, Q. Liu, M. Zhang, Y. Wang, and X. Deng, "Gradient-based low rank method and its application in image inpainting," *Multimedia Tools and Applications*, vol. 77, no. 5, pp. 5969–5993, 2018.
- [27] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 29, no. 3, pp. 463–476, 2007.
- [28] T. Shih, N. Tang, and J.-N. Hwang, "Exemplar-based video inpainting without ghost shadow artifacts by maintaining temporal continuity," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, pp. 347–360, 2009.
- [29] Y. Umeda and K. Arakawa, "Removal of film scratches using exemplar-based inpainting with directional median filter," in *2012 International Symposium on Communications and Information Technologies (ISCIT)*, 2012.
- [30] Z. Xu, Q. Zhang, Z. Cao, and C. Xiao, "Video background completion using motion-guided pixel assignment optimization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 8, pp. 1393–1406, 2015.
- [31] J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, "Temporally coherent completion of dynamic video," *ACM Transactions on Graphics (ToG)*, vol. 35, no. 6, pp. 1–11, 2016.
- [32] A. Ghanbari and M. Soryani, "Contour-based video inpainting," in *2011 7th Iranian Conference on Machine Vision and Image Processing*, 2011.
- [33] A. Xia, Y. Gui, L. Yao, L. Ma, and X. Lin, "Exemplar-based object removal in video using gmm," in *2011 International Conference on Multimedia and Signal Processing*, 2011.
- [34] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Advances in neural information processing systems*, 2012.
- [35] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [36] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [37] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–14, 2017.
- [38] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [39] 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 *Proceedings of the European Conference on Computer Vision*, 2018.
- [40] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [42] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [43] 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 *Proceedings of the European Conference on Computer Vision*, 2018.
- [44] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 DAVIS challenge on video object segmentation," *arXiv:1704.00675*, 2017.
- [45] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [46] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems*, 2017.
- [47] S. Min, X. Chen, Z.-J. Zha, F. Wu, and Y. Zhang, "A two-stream mutual attention network for semi-supervised biomedical segmentation with noisy labels," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.