Deep Transformer based Video Inpainting Empowered by Fast Fourier Tokenization

Jinwoo Kim, Woojae Kim, and Sanghoon Lee, *Senior Member, IEEE*

***Abstract*—Bridging distant space-time interactions is important for high-quality video inpainting with large moving masks. Most existing technologies exploit patch similarities within the frames, or leaverage large-scale training data to fill the hole along spatial and temporal dimensions. Recent works introduce promising Transformer architecture into deep video inpainting to escape from the dominanace of nearby interactions and achieve superior performance than their baselines. However, such methods still struggle to complete larger holes containing complicated scenes. To alleviate this issue, we first employ a fast Fourier convolutions, which cover the frame-wide receptive field, for token representation. Then, the token passes through the seperated spatio-temporal transformer to explicitly moel the long-range context relations and simultaneously complete the missing regions in all input frames. By formulating video inpainting as a directionless sequence-to-sequence prediction task, our model fills visually consistent content, even under conditions such as large missing areas or complex geometries. Furthermore, our spatio-temporal transformer iteratively fills the hole from the boundary enabling it to exploit rich contextual information. We validate the superiority of the proposed model by using standard stationary masks and more realistic moving object masks. Both qualitative and quantitative results show that our model compares favorably against the state-of-the-art algorithms.**

***Index Terms*—Video inpainting, Video completion, Free-form inpainting, Object removal, adversarial learning**

I. Introduction

VIDEO inpainting refers to the task of filling reasonable content with a spatially and temporally coherent appearance into missing regions, conditioned on partially visible video. An effective video inpainting algorithm has been applied to a wide range of real-world applications, including restoration (removing permanent defects such as scratches and dust), video re-touching (removing unwanted objects and watermarks), and stabilization (reducing fluctuated motion and de-flickering). However, high-quality video inpainting is still challenging due to the lack of long-range interactions within the space-time regions.

Early patch-based video inpainting methods fill the masked region by pasting the most similar patch somewhere in the video [1]–[3]. These methods are often time-consuming and have shown limited ability to synthesize non-repetitive and complex regions because they assume there is a hint for the missing parts in the visible regions. Recent learning-based techniques have significantly raised its performance bar by using 3D convolutions and recurrent networks [4]–[6]. These approaches typically aggregate information from nearby frames to fill in the missing regions. The most successful algorithms to date are attention-based modules to transfer long-range relations between visible and invisible regions in video [7], [8]. Despite the significant advances, the main challenge of this task is a requirement of bridging and exploiting visible information into synthesized contexts considering inter-frame and intra-frame relationships.

Recently, Transformers are on the rise and they are now the de-facto standard architecture for language tasks [9]–[11], and lately, start to perform comparably or even better than their convolutional neural networks (CNNs) in a variety of vision benchmarks [12], [13]. Compared to CNN models, the transformer has strong representation capability and is free from inductive bias. By allowing long-term interactions via the dense attention module, some preliminary works demonstrate its capacity in modeling the structure relationships for natural image synthesis and produces natural outputs by optimizing the underlying data distribution [14]–[16].

Inspired by the emerging trend of using transformer architecture for computer vision tasks, we propose a new high-fidelity pluralistic video inpainting method. Specifically, we treat video inpainting as a directionless sequence-to-sequence prediction task that captures short- and long-term interactions within multi-head self-attention mechanisms. However, as discussed in the literatures [17]–[19], transformers are good at capturing long-range interactions on the input tokens, but they are less efficient at capturing fine-grained local dependencies. Convolution layers, on the other hand, are designed to effectively capture local details but require deeper layers for understanding the global context. This implies that transformers and CNNs have their own limitations.

In this paper, our key insight is to bridge the best of both architectures: transformer layers interact global structural dependences and convolution layers refine the local texture contexts using these global structural understandings. However, it remains a challenge to directly apply these transformer models to visual generation task. Particularly, unlike natural language processing (NLP), which treats each word as a vector for token embeddings, it is unclear what token representation shold be good for visual tasks. Therefore, previous studies exploit every pixel or non-overlapping patches ($e.g. 16 \times 16$) as a token representation, but due to the high memory requirements with input length, methods suffer from resolution issue [20]. To mitigate this issue, we adopt convolutional layers to efficiently learn the compositional nature of the masked video frames. Nevertheless, we noticed that popular convolutional architectures might lack a sufficient large receptive field for efficient token representations [13], [14], [16]. To achieve this, we propose a method of token representation based on recently developed fast Fourier convolutions (FFC) [21], [22]. This has a profound influence allowing for the frame-wise receptive

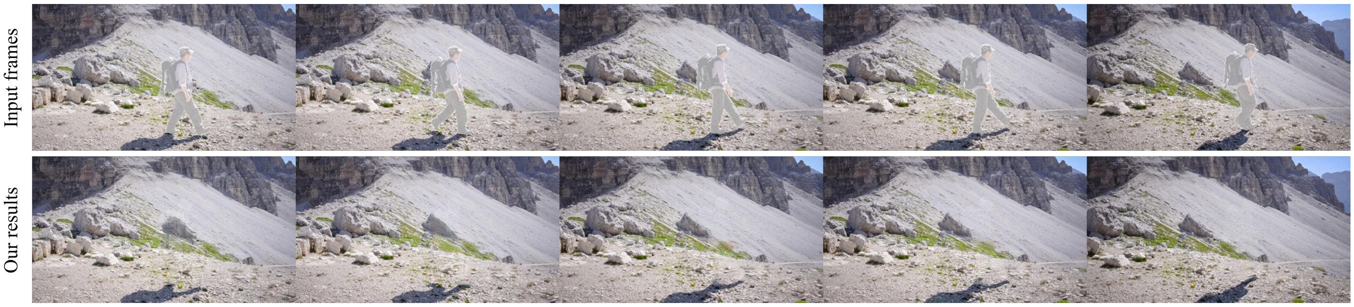

Fig. 1. We propose transformer based video inpainting network with iterative refinement. Our model try to complete the gray regions at the top row and we visualize synthesized frames at the bottom row (please zoom in to see the details).

field that covers an entire frame even in the early layers of the network.

The proposed token representation improves video inpainting performance to a great extent, yet still suffers from the computational complexity of its self-attention which is quadratic to frame length and would be intractable for a transformer on videos. To this end, we propose a spatially and temporally separated transformer backbone that searches coherent tokens from all the frams and completes all input frames at once. Specifically, we decouple the transformers over the space-time volumes to effectively process the large number of spatio-temporal tokens that may be encountered in the video. This design allows the model to synthesis the stationary background texture in the intra-frame and then refine the temporal consistency in the inter-frame. We empirically evaluate this for the several scalable transformer designs. Additionally, to effectively complete the details even for large hole samples, we iteratively refine the token by gradually eroding the hole. Our design recurrently infers and gathers the hole boundary for the encoded feature map. By doing so, our network can exploit richer contextual information for the missing regions at each iteration.

Furthermore, it has been a common understanding that transformers are "data-hungry" models because of inductive bias free design and they require sufficiently large datasets. However, video datasets are relatively small to train. We note that how to train transformer models effectively on smaller video datasets by pre-training with large image datasets. Our training procedure takes advantage of a set of still images to pre-train the proposed network. Figure 1 shows sample result producing satisfactory content in challenging object removal case. Through extensive experiments, we demonstrate that our model outperforms the state-of-the-arts by a significant margin in terms of PSNR, SSIM and VFID. We also verify the effectiveness of the proposed methods through ablation studies. Our contributions can be summarized as follows.

1) We propose a video inpainting network based on recently developed FFCs. The FFCs not only allow the context rich token representations but also refine the local texture details. This significantly enhances network performance.
2) We propose an interweaving spatial-temporal transformer framework to effectively capture global structural dependencies. The hierarchical transformer allows intra- and inter-frame tokens to freely attend to the spatially and temporally coherence feature to restore the global structure.
3) We propose an iterative refinement module to further refine the deeper pixels in the holes by gradually gathering richer contextual information for the missing regions at each step.

The remainder of this article is organized as follows. Section II introduces related work (e.g. image inpainting and video inpainting) to review the latest algorithms. We describe the overall framework including the background of FFCs, Transformer, the architecture of the proposed model, and the training details in Section III. Section IV introduces the database used in the performance evaluation and experimental results. Lastly, concluding remarks are given in Section V.

II. Related Works

A. Image inpainting methods

Traditional works for image inpainting can broadly include either diffusion-based [23]–[25] or patch-based methods [26]–[28]. The former propagates texture from knwon regions to unknown (mising) regions, and works well with small holes but suffers from artifacts and noisy results with larger holes. The latter have focused on matching and copying the nearest neighbor background patches. More recently, many reserchers have utilized large image datastes to produce semantically consistent content by applying learning-based methods. In particular adversarial training can make the inpainted images more realistic [29]–[32]. The context encoder is one of the early attempts to generate resonable results based on feature learning [33]. The follow-up methods improve the visual quality of the inpainted images to handle the free-form mask and adopted a two stage refinement structure (e.g. coarse to fine architectures including edges and structures) [34], [35].

Evolving from these works, numerous studies have tried to use attention layers learns the correlation between background and foreground feature maps to borrow pixels from distant locations [22], [36]. To further refinement, image inpainting methods adopted a recursive hole-filling scheme to cover a large hole. This methods ensure the confidence region from the boundary to the center within the feature spaces [37]. Our work leverages attention and iterative refinement frameworks to the video inpainting task.

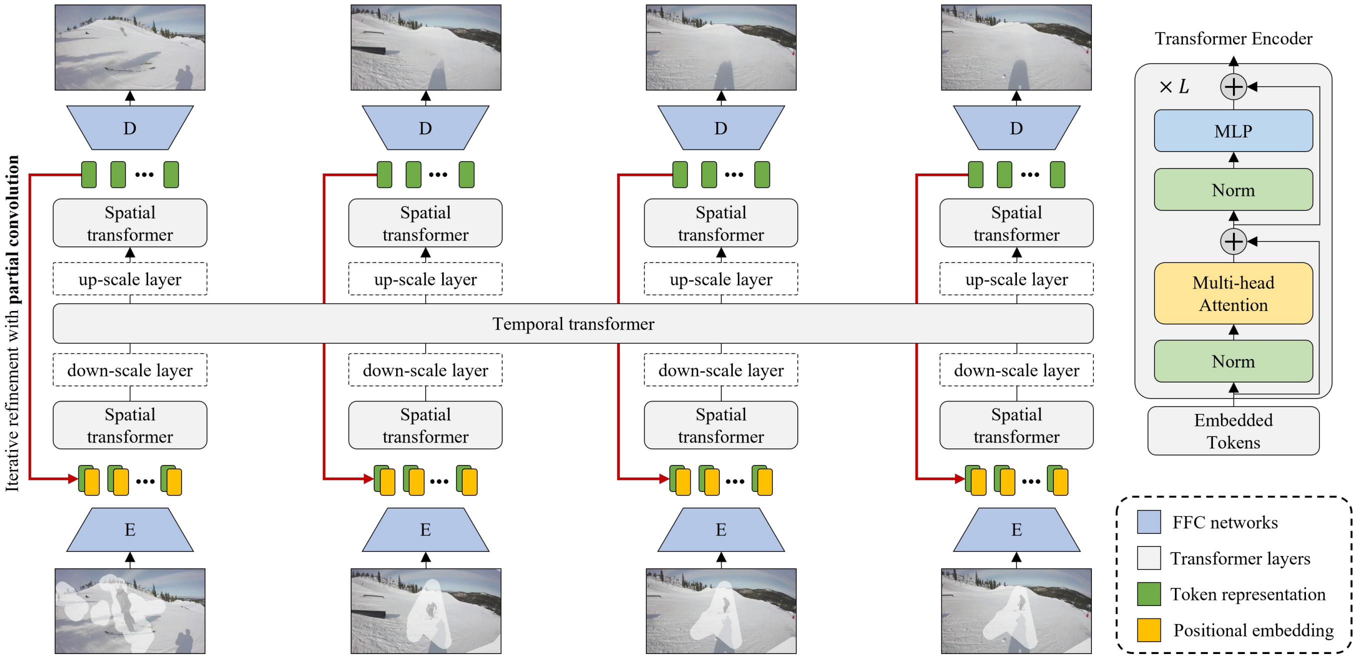

Fig. 2. Overview of the **T**ransformer-based **V**ideo **I**npainting (TVI) architecture. TVI consists of separated spatial and temporal transformer blocks with iterative refinement module. The model first embed multiple-input frame with independent tokens and then simultaneously fill the hole by passing through transformer module for the globally coherent synthesis.

B. Video inpainting methods

Video inpainting not only inherits the challenges faced by the image inpainting task, but it should produce time-consistent content. Early video inpainting methods mainly formulate space-time filling processes as patch-based optimization techniques [26], [38], [39]. These methods complete the holes by borrowing 3D (spatio-temporal) patches as the synthesis unit. In particular, Huang *et al.* propose a non-parametric optimization formulation that combines flow-field estimation and flow-guided patch synthesis [2], [3]. While these methods achieve impressive results, they typically assume stationary motion fields in holes and are often constrained by dynamic camera motion. There is also a computation issue.

More recently, many researchers have utilized large datasets to generate plausible content by applying a deep learning model. The combined 2D and 3D CNNs is one of the early attempts to learn temporal and spatial features [6], but produces blurry results. Inspired by flow-based methods [40], [41], Xu *et al.* explicitly estimate both appearance and optical-flow to facilitate propagating content from potentially distant frames. Kim *et al.* introduce a recurrent network to aggregate temporal features from nearby frames [5]. Chang *et al.* develop free-form video inpainting with 3D gated convolution and temporal PatchGAN [4]. Owing to the limited representation ability to model long-range correspondences, these methods may fail to capture visible content from distant frames.

To alleviate this issue, recent approaches have adopted attention modules and show reasonable performance. Oh *et al.* progressively fill the missing regions from boundary to center with an asymmetric attention for calculating the similarities between the target and reference frames [8]. Zeng *et al.* propose STTN by directly transferring the multi-head self-attention to a video inpainting task and the model simultaneously completes the input frames considering the spatial-temporal similarity [42]. However, STTN brings a huge computation cost. Specifically, applying a single multi-head attention layer to images with pixel resolution of 128×128 with 8 batches still requires more than 32GB of memory, which is generally impractical. Inspired by STTN, Liu *et al.* propose DSTT disentangling the spatial and temporal learning task into 2 sub-tasks [43]. Unlike DSTT works, our method iteratively fills the missing hole in the feature domain where continents tokens are extracted each frame with smaller dimensions to propagate the long-range interaction over the space-time regions. Furthermore, we go one step further to present efficient training of data-hungry transformer architecture.

III. Methods

A. Overview

1) Problem formulation: Let $X_1^T = \{X_1, X_2, ..., X_T\}$ denote a set of corrupted video frames with sequence length T and $M_1^T = \{M_1, M_2, ..., M_T\}$ be the corresponding frame-wise masks. We aim to learn a mapping function that produces reasonable video output which can be expressed as $G : X_1^T \rightarrow \hat{Y}_1^T$, where $\hat{Y}_1^T = \{\hat{Y}_1, \hat{Y}_2, ..., \hat{Y}_T\}$ is the predicted video frames. Such $\hat{Y}_1^T$ can be approximated as close as the target video $Y_1^T = \{Y_1, Y_2, ..., Y_T\}$. To do this, we formulate a video inpainting as a multi-input and multi-output generative task where we estimate the conditional distribution $p(Y_1^T|X_1^T)$.

Specifically, our motivation is that a missing hole in a current frame would probably be revealed both in adjacent and distant frames. The hidden region can be filled from adjacent frames by borrowing texture information when a mask is moving fast. Conversely, the occluded region can be revealed in a distant frame when a mask is large and moving slowly. Therefore, our model takes both adjacent and distant frames as conditions, and then simultaneously fills the missing input frames. Following the Markov assumption [5], [42], we factorize the multiple conditional inputs and corresponding multiple outputs as a product form which is denoted as:

$$p(\hat{Y}_1^T|X_1^T) = \prod_{t=1}^{T} p\big(\hat{Y}_t^{t+T_R}|X_t^{t+T_R}, X_{1,s}^T\big), \tag{1}$$

where $X_t^{t+T_R}$ and $X_{1,s}^T$ denote the adjacent and distant frames, respectively. Following the previous study [42], the distant frames $X_{1,s}^T$ are uniformly sampled from the total frames X_1^T in a sampling rate of s.

2) Network design: To infer spatially and temporally consistent contents, we design a new inference model, called **TVI**, that employs a **T**ransformer-based **V**ideo **I**npainting architecture. The overview of our video inpainting framework is shown in Figure 2. Specifically, TVI consists of a frame-level encoder-decoder and cascaded spatial-temporal transformer. The frame-level encoder captures low-level frame structure leaveraging FFC. Similarly, the frame-level decoder is designed to restore features back to frames. The cascaded spatial-temporal transformer is responsible for capturing long-range interaction to restore the global context. Another important design is a plug-and-play recurrent feature reasoning process to enforce each global structure prediction which infers and gathers the hole boundary for the encoded feature map. In this way, the constraints determining the spatial-temporal contents are progressively strengthened and the model can produce semantically explicit results.

B. Fast Fourier Convolution

1) Background: The conventional fully convolutional models might be insufficient to ensure the large receptive field due to the typically small ($e.g.$3 × 3) convolutional kernels. Thus, these models require deeper layers in the network which has a large memory footprint. In particular, in a video inpainting task with large moving masks, the insufficient receptive field of the generator tends to observe the missing pixels around it. The issue becomes especially pronounced for completing visually coherent contexts.

FFC [21] is the recently proposed approach that explores an efficient ensemble of local and non-local receptive fields in a single unit. This method is based on a channel-wise fast Fourier transform (FFT) [44] that covers global context for all layers by enlarging the image-wide receptive field. FFC contains two inter-connected branches: i) a spatial (or local) branch explits conventional convolutions on a part of input feature channels, and ii) a spectral (or global) branch is operated spectral domain utilizing real FFT to account for global context. Each branches can capture local and global information simultaneously with a different receptive field. The complementary feature aggregation between both branches is performed internally.

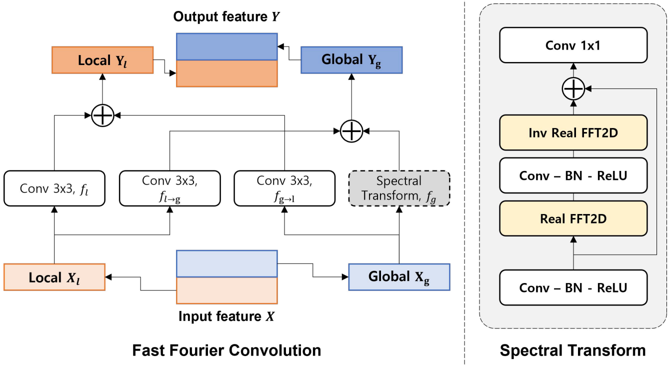

Fig. 3. Illustration of Fast Fourier Convolution (FFC). The token representation is based on FFC which ensures a large receptive field and can avoid meaningless operation on the large hole regions. "$\oplus$" denotes element-sise sum.

Specifically, lex $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$ be the input feature volumes of FFC layer, where H, W, and C represent the spatial resolution (e.g. height and width) and the number of channels, respectively. Then, $\mathbf{X}$ is embedded into two parallel branches by splitting the dimension along the channel axis. The split local and global features are denoted as $\mathbf{X} = \{\mathbf{X}_l, \mathbf{X}_g\}$. Local feature $\mathbf{X}_l \in \mathbb{R}^{H\times W\times(1-\alpha_{in})C}$ learns local details using traditional convolution operation. Global feature $\mathbf{X}_g \in \mathbb{R}^{H\times W\times\alpha_{in}C}$ captures the global context by transforming the spatial domain into the spectral domain using Real FFT. By utilizing half of the spectrum compared to the FFT, real FFT is only applicable to real-valued signals, and likewise, inverse real FFT ensures that the output is real valued. α_{in} represents the percentage of feature channels allocated to the global part which is ranging from 0 to 1. The output features of the local and global branches are aggregated togather and provide final feature volume $\mathbf{Y} = \{\mathbf{Y}_l, \mathbf{Y}_g\}$. The entire procedure within internal FFC can be formulated by

$$\mathbf{Y}_l = \mathbf{Y}_{l\to l} + \mathbf{Y}_{g\to l} = f_l(\mathbf{X}_l) + f_{g\to l}(\mathbf{X}_g) \tag{2}$$

$$\mathbf{Y}_g = \mathbf{Y}_{g\to g} + \mathbf{Y}_{l\to g} = f_g(\mathbf{X}_g) + f_{l\to g}(\mathbf{X}_l), \tag{3}$$

where f_l, $f_{g\to l}$, and $f_{l\to g}$ is convilution operation with 3 × 3 kernel shape, and $f_{l\to g}$ represents spectral transformer. The architectural design follows LaMa [22] that applies a single Fourier unit as depicted in Figure 3 right.

2) FFC-based encoder: To leverage the highly expressive transformer for synthesis, we first need to represent eaxh fixed-size frame $(432 \times 245 \times 3)$ as an independent token. However, building on individual pixels as a token is not feasible representations to train the transformer due to the increasing sequence length $(298,080$ tokens for each frame). To feed masked frames X_1^T into the transformer with feasible sequence lengths, we first incorporate the representation abilities of FFCs inspired by the ideas of neural discrete representation learing [14], [16]. Specifically, the frame-level encoder is designed by stacking several FFC layers with down-sampling to capture both global context and local details from the early layers, which is crucial for compact token representation. After that, we obtain spatial features along the temporal index T_R with size $H \times W \times C \times T_R$, and then flatten each spatial feature to a 1D sequence of $(HW \times C) \times T_R$. In our implementation, H, W, C and T_R are set to 30, 54, 512 and 4, respectively. We empirically show our superiority quantitatively and qualitatively in Section IV-D.

3) FFC-based decoder: After token representation, we feed the obtained token into an proposed transformer for fully aggregating information. These processed tokens then are mapped to the target frame through a frame-level decoder. We choose FFC layers as our frame-level decoder to guarantee photorealistic synthesis by gradually up-sampling with groups of dilated convolution. [29]. Due to the image-wide receptive field that covers the entire image, FFC allow superior performance compared to the recent CNN based architecture.

C. Cascaded spatial and temporal transformer

1) Background: We choose the transformer encoder [13] as our basic block. Here, we briefly review the functionality of the transformer. The main operation performed in this layer is self-attention, and it is computed on a sequence of tokens. As depicted in Figure 2 right, the transformer encoder consists of alternating layers of multi-head self-attention (MSA) which is responsible for capturing long-range dependencies and multi-layer perceptron (MLP) blocks with GELU non-linearity. The Layernorm (LN) is applied before both of the two parts and each block employees residual connection. These are denoted as:

$$\mathbf{z}_0 = \left[\mathbf{x}^1; \mathbf{x}^2; ...; \mathbf{x}^M\right] + \mathbf{E}_{pos}, \tag{4}$$

$$\mathbf{z}_l^{'} = \mathbf{MSA}\left(\mathbf{LN}\left(\mathbf{z}_{l-1}\right)\right) + \mathbf{z}_{l-1}, \tag{5}$$

$$\mathbf{z}_l = \mathbf{MLP}\left(\mathbf{LN}\left(\mathbf{z}_l^{'}\right)\right) + \mathbf{z}_l^{'}, \tag{6}$$

where $\mathbf{z} \in \mathbb{R}^{M \times C}$ is the 1D sequence of M tokens $\mathbf{x}$ with C dimensions, and $\mathbf{E}_{pos} \in \mathbb{R}^{M \times C}$ is the position embeddings.

2) Transformer for spatio-temporal interaction: We propose a transformer-based architecture to search coherent contents by taking all the represented tokens, and we will analyze this architecture decision in Appendix B. As illustrated in Fig. 2, our model consists of three separate transformer encoders in series. Similar to BERT [45], the spatial transformer encoder takes embedding tokens as inputs and calculates the correspondence between each token from the same temporal index. A representation for each temporal index is denoted as $\mathbf{x}_s^m \in \mathbb{R}^{HW \times C}$, where $m = 1, 2, ..., T_R$.

To calculate temporal relationships, the tokens from spatial transformer encoder are reshaped along the temporal dimension, $HW \times C \times T_R \rightarrow (HWT_R) \times C$. However, the sequence length dramatically increases along with the number of the input frames T_R, leading to a computational burden. To mitigate the sequence length issue, we add the down-scale and up-scale layers before and after the temporal transformer encoder. Specifically, the down-scale layer reshapes the 1D sequence of token embedding back to a 2D feature map $\mathbf{x}_s^m \in \mathbb{R}^{H \times W \times C}$ and then adopts the stacked convolutions with the down-sampling module $\mathbf{x}_s^m \downarrow \in \mathbb{R}^{H/2 \times W/2 \times C}$. After that, the 2D feature map is again reshaped into the 1D sequence of embedding tokens where the sequence number becomes $\mathbf{x}_t \in \mathbb{R}^{(\frac{H}{2}\frac{W}{2}T_R) \times C}$. Then, the temporal transformer encoder takes the temporally grouped token $\mathbf{x}_t$ and calculates the correspondence between each token, recursively. Similarly, the up-scale layer is designed to reshape the temporally calculated token back to $(HW \times C) \times T_R$ dimensions. Finally, spatial transformer encoder is adopted once more to further improve the synthesis quality.

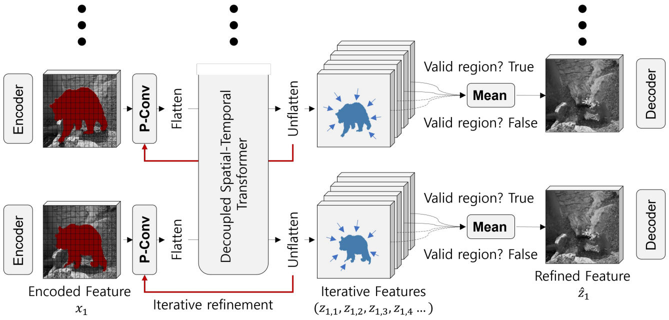

Fig. 4. Illustration of the iterative refinement procedure. The area identification process is performed by partial convolution and the hole region gradually decreases during several times reasoning (blue arrows). After iterative refinements, collected feature maps are adaptively merged considering the valid region of the mask.

3) Inference via iterative refinement:: To further enhance the potential quality, our model iteratively refines the intermediate features through the transformer block by gradually enhancing the internal contents. Unlike existing iterative methods [8], the proposed model performs this refinement in the encoded feature space. By doing so, our model not only reuses the parameters to deliver a much lighter model but also ensures superior performance.

In each interactive process, partial convolution [46] is a basic module used to identify the area to be updated. The operation updates the mask and renormalizes the feature map after the convolution calculation. Let $\mathbf{W}$ denote the convolutional kernel and $\mathbf{b}$ be the corresponding bias. The feature map $\mathbf{x}^*$ computed by the partial convolution layer can be expressed as:

$$\mathbf{x}^* = \begin{cases} \mathbf{W}^T\left(\mathbf{x} \odot \mathbf{m}\right) \frac{\text{if sum}(\mathbf{1})}{\text{sum}(\mathbf{m})} + \mathbf{b}, & \text{sum}(\mathbf{m}) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where $\mathbf{x}$ and $\mathbf{m}$ are the feature values for the current convolution window and the corresponding binary mask, respectively. Similarly, the updated mask value can be expressed as:

$$\mathbf{m}^* = \begin{cases} 1, & \text{if sum}(\mathbf{m}) > 0 \\ 0. & \text{otherwise} \end{cases} \tag{8}$$

Given the equations above, we are able to receive new masks whose holes are smaller after each partial convolution layer.

After several refinement processes through transformer, the intermediate features are merged to avoid gradient vanishing problems as discussed in previous studies [37]. Instead of passing the last features directly to the decoder, we employ an adaptive merging scheme that normalizes the value to the newly completed regions [37]. Let $\mathbf{z}_{m,n}$ denote n^{th} iteration features calculated along the temporal index $m = 1, 2, ..., T_R$. The value at the refined feature map $\hat{\mathbf{z}}_{m,n}$ is defined as:

$$\hat{\mathbf{z}}_m = \sum_{n=1}^{N} \frac{\mathbf{z}_{m,n}}{\mathbf{m}^*_{m,n}}, \tag{9}$$

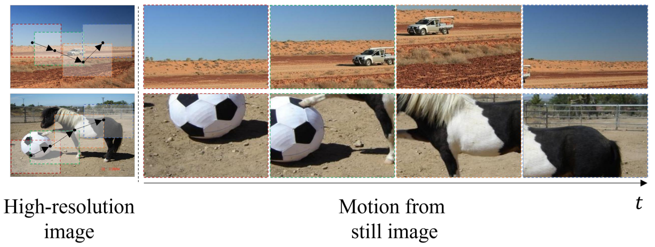

Fig. 5. Examples of cropped images (right) from the high-resolution image (left). Cropped images imitate small to large motions.

where N is the iterative number. This allows the model to merge the arbitrary number of feature maps, ensuring the potential quality of synthesis. The details of iterative refinement pipeline of our module is illustrated in Figure 4.

D. Training

1) Loss function: The loss function is designed to capture pixel-wise reconstruction accuracy, perceptual similarity, and temporal consistency. To this end, we minimize the $L1$ distance between the generated and the ground-truth frames for ensuring pixel-wise reconstruction. The pixel losses are defined as follow:

$$\mathcal{L}_{hole} = \left\| (1 - M_1^T) \odot \left(\hat{Y}_1^T - Y_1^T\right) \right\|, \tag{10}$$

$$\mathcal{L}_{valid} = \left\| M_1^T \odot \left(\hat{Y}_1^T - Y_1^T\right) \right\|. \tag{11}$$

We also include structural similarity index measure (SSIM), which is widely utilized as a perceptually motivated loss [47]. This is defined as follow:

$$\mathcal{L}_{SSIM} = \sum_{t=1}^{T} \mathrm{SSIM}\big(\hat{Y}_t, Y_t\big). \tag{12}$$

Furthermore, to preserve temporal consistency, we adopt a Temporal Patch GAN as our discriminator [42]. We do not modify the discriminator architecture or design the loss function in any way. Such an adversarial loss leads to both plausible and coherent results in video inpainting. The optimization function for the discriminator is defined as follow:

$$\begin{aligned}\mathcal{L}_D = & E_{x\sim P_{Y_1^T}(x)}\big[\mathrm{ReLU}\big(1 - D(x)\big)\big] + \\ & E_{x\sim P_{\hat{Y}_1^T}(x)}\big[\mathrm{ReLU}\big(1 + D(x)\big)\big].\end{aligned} \tag{13}$$

Then, the corresponding adversarial loss for TVI are as follows:

$$\mathcal{L}_{adv} = -E_{z\sim P_{\hat{Y}_1^T}}\big[D(z)\big]. \tag{14}$$

Finally, the overall loss function is concluded as below:

$$\mathcal{L} = \lambda_{hole}\cdot\mathcal{L}_{hole} + \lambda_{valid}\cdot\mathcal{L}_{valid} + \lambda_{SSIM}\cdot\mathcal{L}_{SSIM} + \lambda_{adv}\mathcal{L}_{adv}, \tag{15}$$

where the hyperparameters are determined empirically (i.e., λ_{hole}, λ_{hole}, and λ_{hole} are set to 1 and λ_{adv} is 0.1).

```
Algorithm 1: Training of our proposed network
  Inputs  : $X_{1:T} : \{X_1, ..., X_T\}$, Corrupted frames;
            $M_{1:T} : \{M_1, ..., M_T\}$, Frame-wise masks;
  Outputs: $\hat{Y}_{1:T} : \{\hat{Y}_1, ..., \hat{Y}_T\}$, Outputs of the TVI;

  initialization;
  $\mathbf{x}_{1:T} \leftarrow$ FFC Encoder($X_1^T, M_1^T$);
  $\mathbf{z}_{1:T} \leftarrow$ PositionalEncoding($\mathbf{x}_1^T$);
  $\mathbf{m}_{1:T} \leftarrow$ DownSampling($M_{1:T}$);
  $i \leftarrow 0$;
  while $i$ smaller than $N$ do
      $\mathbf{z}_{1:T}^{i+1}, \mathbf{m}_{1:T}^{i+1} \leftarrow$ PartialConv($\mathbf{z}_{1:T}^{i}, \mathbf{m}_{1:T}^{i}$);
      $\mathbf{z}_{1:T}^{i+1} \leftarrow$ SpatialTemporalTransformer($\mathbf{z}_{1:T}^{i+1}$);
      FeatureGroup $\leftarrow$ FeatureGroup + $\{\mathbf{z}_{1:T}^{i+1}\}$;
      $i \leftarrow i + 1$;
  end
  $\mathbf{x}_{1:T}^{merged} \leftarrow$ FeatureMerge(FeatureGroup);
  $\hat{Y}_{1:T} \leftarrow$ FFCDecoder($\mathbf{x}_{1:T}^{merged}$);
  Updating the TVI with loss $\mathcal{L}$;
```

2) Pre-training from image dataset: Transformer-based models are known to "data-hungry" architectures which may be effective when large training datasets are available. However, the video datasets are relatively small to train, suggesting that our model can be easily biased by a limited set of training samples. To mitigate this issue, our training procedure takes advantage of a set of still images to pre-train TVI.

Specifically, we employ large-scale image datasets from high-resolution Places2 [48] to train our model by cropping the single image considering the motion components. Inspired by optical flow methods [49] which assume motion varies smoothly almost everywhere in the real-world video, we generate motion from a still image as follow:

$$\begin{aligned} c_x^{m+1} &= c_x^m + w\cdot\Delta x, \\ c_y^{m+1} &= c_y^m + h\cdot\Delta y, \end{aligned} \tag{16}$$

where (c_x^m, c_y^m) is the center point of the cropped images, and w and h are the width and height, respectively. The differential terms Δx and Δy are randomly sampled variables from the zero-centered normal distribution to capture position change. Figure 5 shows examples of cropped images to pre-train our model.

E. Implementation Details

Our FFC-based encoder and decoder layers are ispired by ResNet architecture [50] replacing CNN layers in residual block with FFC layers. In our model, we use 3 donwsampling blocks and 3 upsampling blocks, respectively. The details of our FFC-based encoder and decoder models in the TVI is described in Table I. Our transformer model is identical to the ViT architecture and we vary its capacity mainly through staking the amount of layers. We discuss transformer capacity quantitatively in Section IV-D. We choose a Temporal PatchGAN (T-PatchGAN) [4] as our discriminaotr. T-PatchGAN consists of six layers of 3D convolution layers. The module

TABLE I

High-level architecture of the encoder and decoder of our TVI. The design of the networks follows the architecture presented in []. Note that $h = \frac{H}{2^m}$ and $w = \frac{W}{2^m}$.

Encoder	Decoder
$x \in \mathbb{R}^{H \times W \times C}$	$\hat{z}_m \in \mathbb{R}^{h \times w \times d_{model}}$
Conv2D $\rightarrow \mathbb{R}^{H \times W \times C'}$	Conv2D $\rightarrow \mathbb{R}^{h \times w \times C''}$
$m \times$ {Residual Block with FFC, Downsample Block} $\rightarrow \mathbb{R}^{h \times w \times C''}$	Non-Local Block $\rightarrow \mathbb{R}^{h \times w \times C''}$
Residual Block $\rightarrow \mathbb{R}^{h \times w \times C''}$	Residual Block $\rightarrow \mathbb{R}^{h \times w \times C''}$
Non-Local Block $\rightarrow \mathbb{R}^{h \times w \times C''}$	$m \times$ {Residual Block with FFC, Upsample Block} $\rightarrow \mathbb{R}^{H \times W \times C'}$
GroupNorm, Swish, Conv2D $\rightarrow \mathbb{R}^{h \times w \times d_{model}}$	GroupNorm, Swish, Conv2D $\rightarrow \mathbb{R}^{H \times W \times C}$

performs classification whether each spatial and temporal feature is real or fake as in the standard GAN setting. Such an adversarial training procedure ensures TVI to focus more on the spatial details and the temporal coherence of real videos [4], [51]. Furthermore, we manually choose the recurrence number N to be 8 transformer module to simplify training. The pipeline of the network is described in Algorithm 1.

Frames with a resolution 432×240 are utilized to train the proposed model. The color values of all frames are linearly scaled to $[-1, 1]$ during all experiments. Before the training procedure, we initialize all weights of the network using the normalized distribution $\mathcal{N}(0, 1)$. We conduct the optimization using the Adam optimizer [52] with $(\beta_1, \beta_2) = (0.0, 0.99)$ for both TVI and discriminator. We set fixed learning rate to $\lambda = 1e^{-4}$. The spectral normalization (SN) [53] is used to stabilize our model by scaling down the weight metrics with their largest singular values. Our model was trained with a batch-size of at least 2 on a GPU with 128GB VRAM, but we generally train on more than 8 GPUs with an accumulated VRAM 96GB. If hardware permits, 16-bit precision training is enabled.

IV. Experiments

In this section, we first introduces the datasets used to validate the model and then describes the training details for each dataset to reproduce the results. To evaluate our approach, we provide quantitative and qualitative analysis comparing with recent video inpainting methods, as well as a user study. Finally, we ablate our model with various baseline components and provide additional results. Our code will be available in:.

A. Datasets

Comparisons are conducted on two commonly utilized datasets adopted in studies on video inpainting: Youtube-VOS [55] and DAVIS [56]. Youtube-VOS is composed of $4,453$ videos with various scenes where the train/validation/test split is divided as 3471, 474, and 508. We follow the original dataset split ans show experimental results on the test set for Youtube-VOS. The average video length in Youtube-VOS is around 150 frames. DAVIS dataset contains 150 high-quality videos of dynamic camera and foreground motions. Following the previous evaluation protocol [41], we use the 60 sequences for training and 90 sequences for testing. We also exploit high-resolution Places2 [48], which is a large-scale image dataset suitable for natural synthesis tasks, to pre-train TVI that are data-hungry model due to the inductive bias free design. We also exploit previous image corrupted methods [54] to simultate real-world application by using three types of free-form masks, including moving object-like mask, moving curve mask, and stationary mask.

Furthermore, we use slightly different training strategies according to datasets. Since the training video dataset is limited, we first train our model leveraging high-resolution Places2 dataset. We only train the generator with appearance loss term for 300 epochs during pre-training procedure. After then, we add the discriminator with adversarial loss to fine-tune the proposed model both on the Youtube-VOS and DAVIS datasets. The model is further fine-tuned 200 epochs.

B. Baselines and Evaluation Metrics

We compare our proposed model with four existing deep-learning based video inpainting methods. We choose the OPN [8], CPN [7], FGVC [41], STTN [42] and PTFAN [54] for comparisons. These models are re-trained until convergence following the same experimental settings proposed in each study. The details of baselines are listed as below:

- **OPN**: adopts iterative refinement and embeds attention modules in the intermediate layers.
- **CPN**: can compute affine matrices by combining reference frame features based on similarity between images.
- **FGVC**: alleviates the limitations of existing flow-based video completion algorithms by leaveraging flow-edge, non-local flow, seamless blending modules.
- **STTN**: learns joint spatial and temporal attention modules using multi-scale patch-based video frame representations.
- **PTFAN**: progressively enriches current frame features with neighbouring frames using optical flow, which aligns temporal feature within the network.

We conduct quantitative comparison in terms of peak signal-to-noise ratio (PSNR), structure similarity index (SSIM), and Video Fréchet Inception Distance (VFID) [57]. The first two metrics, e.g., PSNR and SSIM, assume pixel-wise independency, which may mark favorable scores to perceptually unreasonable results. Therefore, we employ the VFID, which calculates the distance between features using a pre-trained I3D model [58]. Note that these statistics rely on the completed video, which mostly consists of the original parts. Therefore,

TABLE II

QUANTITATIVE COMPARISONS ON TWO DATASETS USING OBJECT MASK, CURVE MASK AND STATIONARY MASK. THE BEST MEASURE ARE IN BOLD. † LOWER VALUE IS BETTER. ∗ HIGHER VALUE IS BETTER.

	Youtube-Vos									DAVIS								
	Object mask			Curve mask			Stationary mask			Object mask			Curve mask			Stationary mask		
	PSNR*	SSIM*	VFID†	PSNR*	SSIM*	VFID†	PSNR*	SSIM*	VFID†	PSNR*	SSIM*	VFID†	PSNR*	SSIM*	VFID†	PSNR*	SSIM*	VFID†
OPN	33.53	0.8844	0.7618	34.16	0.9125	0.6602	36.15	0.9540	0.4004	32.91	0.8635	0.3664	33.78	0.9105	0.2701	36.33	0.9596	0.1281
CPN	33.18	0.8764	0.8257	32.88	0.8676	0.8841	35.86	0.9485	0.4606	32.60	0.8452	0.4331	32.47	0.8496	0.4802	36.55	0.9547	0.1637
FGVC	33.13	0.8832	0.7640	34.14	0.9212	0.640	35.09	0.9422	0.4017	31.95	0.8323	0.4010	32.84	0.8841	0.3432	33.92	0.9212	0.1734
STTN	34.86	0.9047	0.7276	36.07	0.9411	0.6136	39.60	0.9716	0.3132	33.60	0.8708	0.3831	34.83	0.9251	0.2882	38.78	0.9690	**0.1197**
PTFAN	35.48	0.9160	0.6129	37.43	0.9566	0.3661	**41.41**	0.9738	0.2893	34.23	0.8798	0.3526	**36.54**	**0.9508**	**0.1933**	42.05	0.9737	0.1303
TVI	**35.57**	**0.9166**	**0.6072**	**37.88**	**0.9631**	**0.3650**	39.87	**0.9741**	**0.2869**	**34.84**	**0.8832**	**0.3403**	35.93	0.9371	0.2207	**42.21**	**0.9739**	0.1208

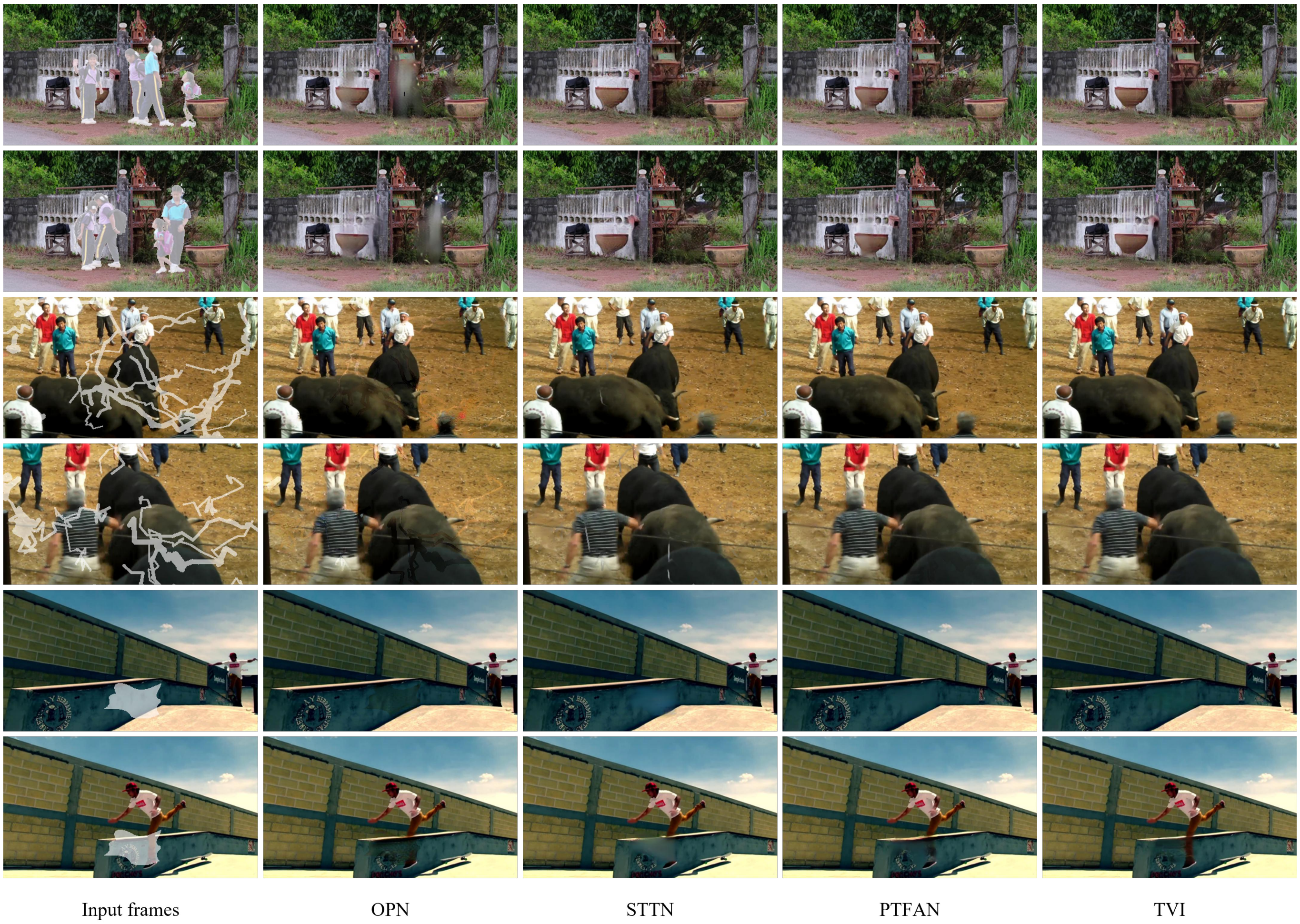

Fig. 6. Qualitative comparisons of our methods with OPN [8], STTN [42], and PTFAN [54]. Our model generates globally coherent contents than other benchmarks.

our reported VFID scores are lower than those of the other generative models.

C. Performance Evaluation

1) Quantitative Comparison: We report quantitative results for filling different three masks (e.g. object, curve, and stationary masks) on Youtube-VOS and DAVIS in Table II. The results show that our model perform better video completion performance over SOTA algorithms [7], [8], [41], [42], [54] on object mask for all evaluation metrics. On the other hand, our method shows that potentially superior or competitive performance in other masks through all evaluations. These resutls indicate that our proposed module is critical to improve the visual quality of the inpainted videos. Furthermore, we also report running time analysis to compare the detailed efficiency of our model in right of Table III. Our model achieves the fewest FLOPs and highest FPS, showing the high efficiency in video inpainting.

2) Qualitative Comparison: To demonstrate the superiority of the proposed method, we report the notable results to address the short-range and long-range interactions in video inpainting studies. Figure 6 illustrates video inpainting samples for object removal, curve mask, and stationary mask corruption, respectively. In all these cases, our inpainting results have

TABLE III
USER STUDY AND MODEL EFFICIENCY ON DIFFERENT METHODS. B IS SHORT OF BILLION.

	TVI WinRate (User study)			Model efficiency	
	Object mask	Curve mask	Stationary mask	FLOPs	FPS
OPN	81.30%	64.32%	73.58%	367B	12.7
STTN	69.48%	56.58%	66.28%	233B	24.3
PTFAN	56.18%	53.79%	59.32%	146B	33.8
TVI	-	-	-	**96B**	**37.9**

significantly plausible content with spatially and temporally coherence for all mask types. Especially, our model could synthesize sharp and clear appearance in object removal tasks by preserving the background textures in invisible or occluded regions. More aditional results are available in Section IV-D.

3) User study: To alleviate a possible bias of the selected evaluation metrics, we further perform a user study to evaluate the visual quality of our model where OPN, STTN, PTFAN are chosen as strong comparison baselines. We randomly sample 20 videos on the DAVIS test split and corrupt sampled videos using object, curve, and stationary masks. Then, we complete corrupted videos through baseline models. Paired comparisons between TVI and the baselines were conducted with the same set of videos. 23 subjects participated in the user study. Each subject is asked to select one between our results and one randomly selected counterpart for the question of "which video is more plausible or visually natural?". Our method obtains the majority of votes compared to all baselines. Specifically, the win-rate of TVI against the baselines are as follows: OPN (81.3%), STTN (69.48%), and PTFAN (56.18%) on the object mask. This implies that the video inpainting produced by our method is more preferable and less detectable compared to other baselines.

D. Ablation Studies

In this section, we wolud like to examine the effectiveness of our contributions separately. Here, we mainly discuss the power of Fourier convolutions, role of the transformer blocks, iterative refinement, and effects of pre-training from image dataset.

1) The power of Fourier convolution: Fourier convolutions defined on a periodic convolution are fully differentiable and are easy to plug in and plug out conventional convolutions. Due to the comprehensive receptive field that covers the entire frame on the spectral domain, Fourier convolutions encourage the network to figure out the global context from the beginning of the layer. This is crucial for our video inpainting framework that represents each frame as consecutive tokens because each token is fragile to large and moving masks.

To demonstrate the power of Fourier convolution, we conducted experiments by ablating the token representation methods using recently discussed vision transformer works [13], [14], [16]. As shown in Figure 7, we split each frame as a set of fixed patches and flattens each patch to represent a token following the VIT [13]. As shown in Table IV and Figure 8, the results achieve temporally coherence appearances but reveal blurry texture quantitatively and qualitatively. We conducted token representation following the VQGAN [14] that first encodes the image using conventional convolution layers and then quantizes its feature as a token through the learnable dictionary. The outputs show visually plausible but still be in trouble reconstructing fine details. To further compare token representation method, we exploit the restrictive CNN [16] that ensures each token to represent individual information without being entangled with neighboring pixels. This method can achieve relatively superior quantitative and qualitative results. However, some details are stilll poor.

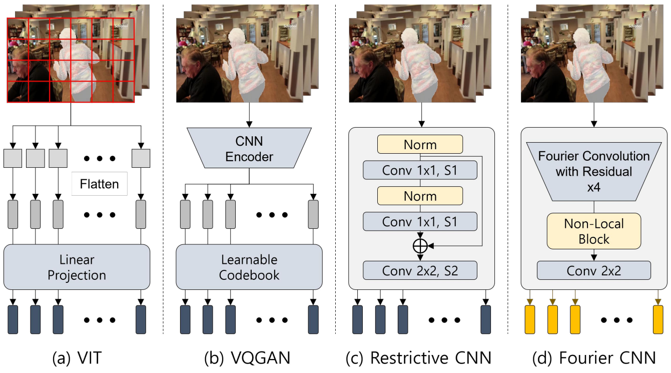

Fig. 7. Visualization of token representation. (a) Patch based token representation [13]. (b) Discrete feature to token [14]. (c) Restricted receptive field feature to token [16]. (d) Fast Fourier convolution based token representation.

TABLE IV
COMPARISONS WITH DIFFERENT CONFIGURATIONS OF VIDEO INPAINTING ARCHITECTURES FOR THE OBJECT REMOVAL TASK.

Method	Youtube-Vos			DAVIS		
	PSNR	SSIM	VFID	PSNR	SSIM	VFID
A Traditional Convolution	30.08	0.7718	0.8921	25.83	0.7859	0.5836
B + VIT	32.52	0.7732	0.8639	26.02	0.7937	0.5517
C + VQGAN	29.17	0.7950	0.8595	26.69	0.8268	0.5562
D + Restrictive CNN	31.29	0.8352	0.8525	27.58	0.8314	0.5419
E + FourierCNN	33.58	0.8421	0.8184	29.81	0.8381	0.4602
F + Spatial Transformer	33.62	0.8780	0.7709	31.80	0.8427	0.4153
G + Temporal Transformer	34.81	0.8917	0.7490	32.47	0.8649	0.3980
H + Decoupled Transformer	34.96	0.9158	0.6138	33.81	0.8733	0.3764
I + Pre-training from image dataset	**35.57**	**0.9166**	**0.6072**	**34.84**	**0.8832**	**0.3403**

We believe this is due to the small receptive field for token representation. Unlike other low-level vision tasks (e.g. style transfer, color transfer, super-resolution, etc.), in the video inpainting task, a significant area of each frame is corrupted by the mask which leads to missing information in that region. Therefore, a small receptive field for independent token representation can lead to rather useless tokens in video inpainting task. Furthermore, transformer approaches are effective in modeling non-local interactions, but they are less efficient at capturing fine-grained local information. This implies fine-grained token representation is important in video inpainting tasks. In contrast previous token representations [13], [14], [16], our Fourier convolution-based token representation covers global context to efficiently conquer fine-grained token representation. As shown in Table IV and Figure 8, our results show impressive improvements quantitatively and qualitatively.

2) Effectiveness of the seperated spatial and temporal transformer: Before we decide our architecture configuration,

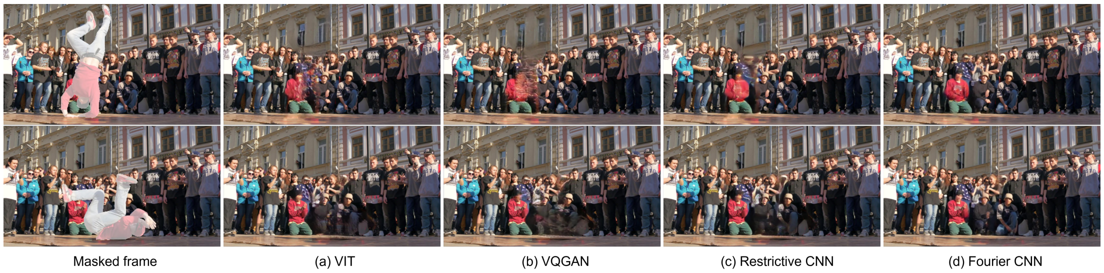

Fig. 8. Qualitative comparisons of different token representations.

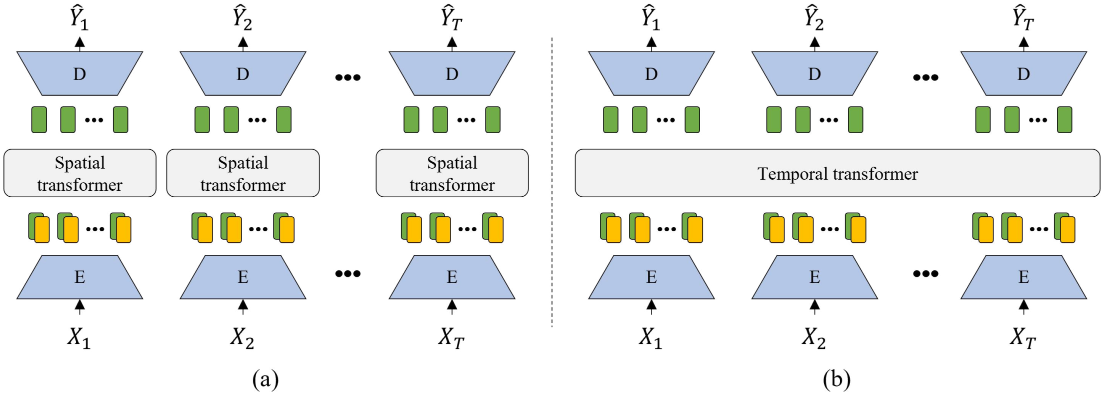

Fig. 9. Architecture selections. (a) consists of a spatial transformer that interacts with frame-wise information. (b) consists of a temporal transformer that interacts with sequential frame information simultaneously.

we rigorously validated two plausible baseline models, as shown in Figure 9 (a) and (b). The two baseline networks differ in their role of interaction range. Specifically, the spatial transformer in Figure 9 (a) takes as input a sequence of each frame token with positional embeddings and interacts relationship within a frame level. Therefore, this network reconstructs the frame without considering temporal dependences. In contrast, the temporal transformer in Figure IV (b) extends local interactions to global interactions. To do this, the network takes a sequence of entire frame tokens as inputs and calculate dot-product attention. In contrast to these models, our model capture short- and long-range visual dependencies separately.

Table IV shows quantitative performances under three different network configurations which are F, G, and H. We found that configuration I outperforms other architecture for all metrics with significant margins. This is because the ability to capture spatial and temporal coherence features is superior to coupled interaction manner. Note that configuration F and G were stacked with the same number of transformer layers to avoid performance differences depending on the depth of the layer. This can be interpreted that the process of finding a spatial dependency is not only memory efficient by avoiding long-range interactions but also works well at restoring details.

3) Effectiveness of the iterative refinement: One of the core contributions is the iterative refinement module which gradually increases the visible region at the feature levels. Here, we mainly illustrate the influences of the iterative refinement process. The results on DAVIS dataset corresponding to different iteration N are given in Figure 10 below. These quantitative scores are recorded after same training iterations. Above all, this ablation study reveal that our methods is robust to the change of this hyper-parameter. These results also show improved performance with the increasing iteraction numbers. However, when the number of iteractions exceeds 6, the performance gain decreases significantly. Figure 10 shows the detail results after iterative refinement is plugged in.

4) Effectiveness of pre-training from image dataset: To effectively learn the short- and long-term dependency, lots of data are required due to the inductive-free design of the transformer blocks. To overcome this difficulty, we leverage large image datasets and pre-train the proposed model. The configuration F in Table IV shows the quantitative results. The TVI model with pre-training procedure recorded improved PSNR (34.84), SSIM (0.8832), and VFID (0.3403) on DAVIS dataset.

5) Additional visual result: In Figures 11, 12, and 13, we show additional examples on DAVIS and Youtube-VOS datasets that were degraded by object, curve, stationary mask types, respectively.

V. Conclusion

In this paper, we propose transformer-based video inpainting to bridge distant space-time visual dependency. To this end, we first extract tokens from sequential frames borrowing the FFC's representation ability. Then, these tokens interact with separated spatial and temporal transformers that enable inter-frame completion first and then refine intra-frame consistency.

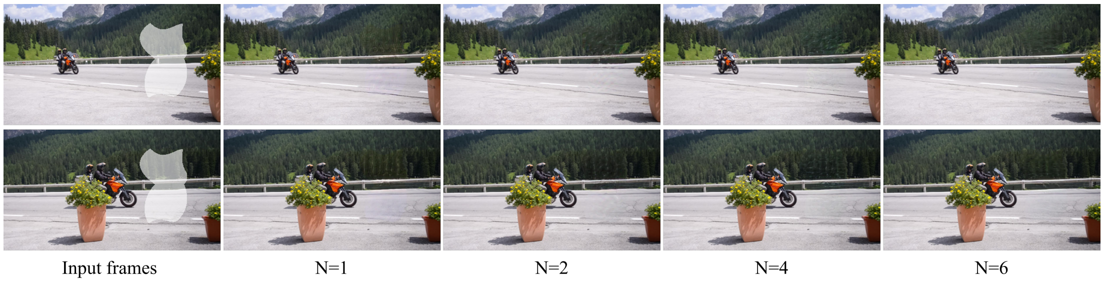

Fig. 10. Comparison results for different iteration numbers.

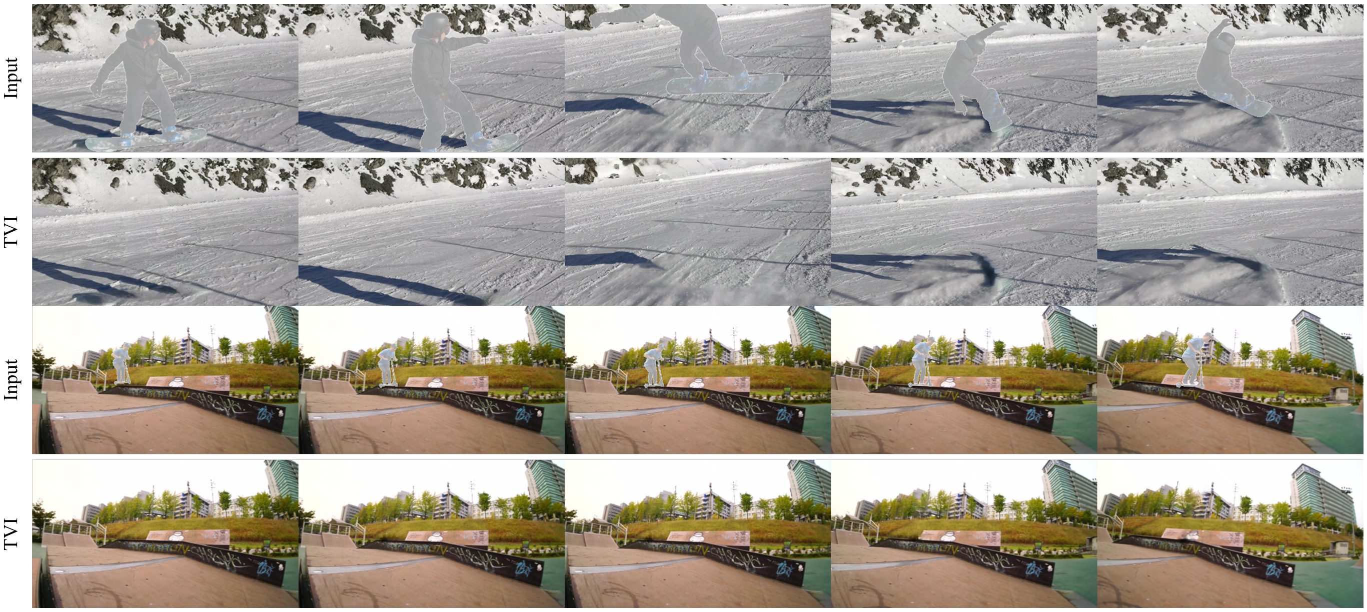

Fig. 11. Additional results on DAVIS dataset for object mask type.

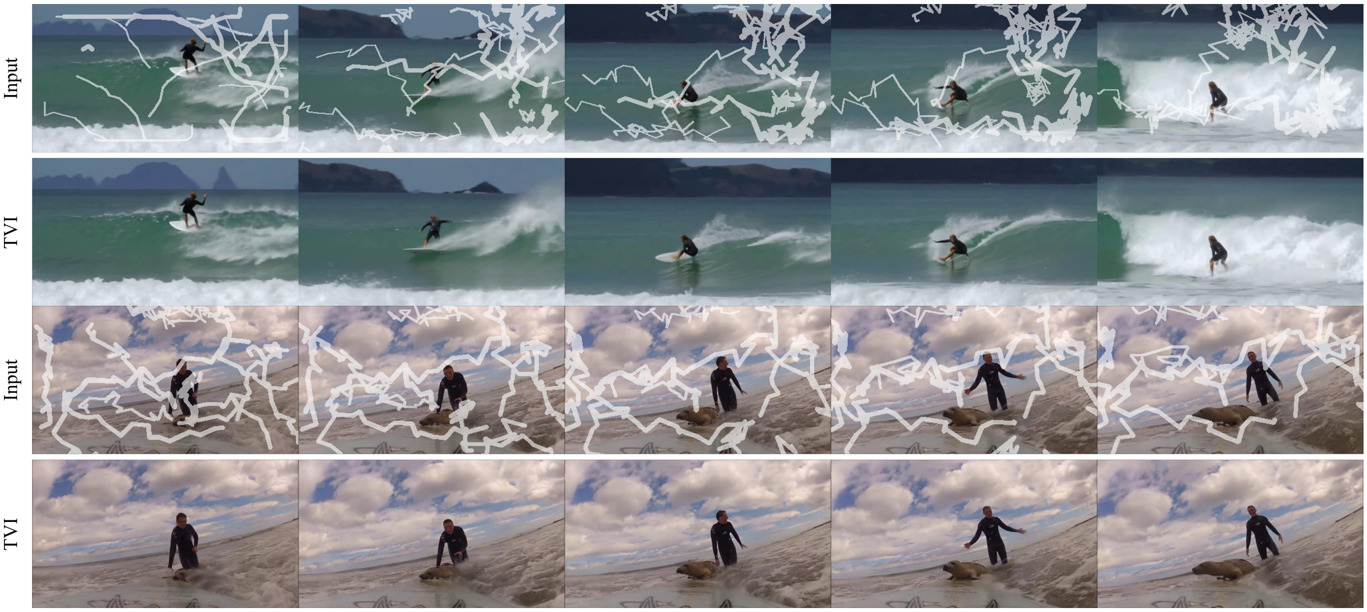

Fig. 12. Additional results on Youtube-VOS dataset for curve mask type.

Our detailed analyses validate that the proposed architectural design achieves effective modeling connections between distant frames. Furthermore, extensive experiments demonstrate that our method is superior to previous video inpainting

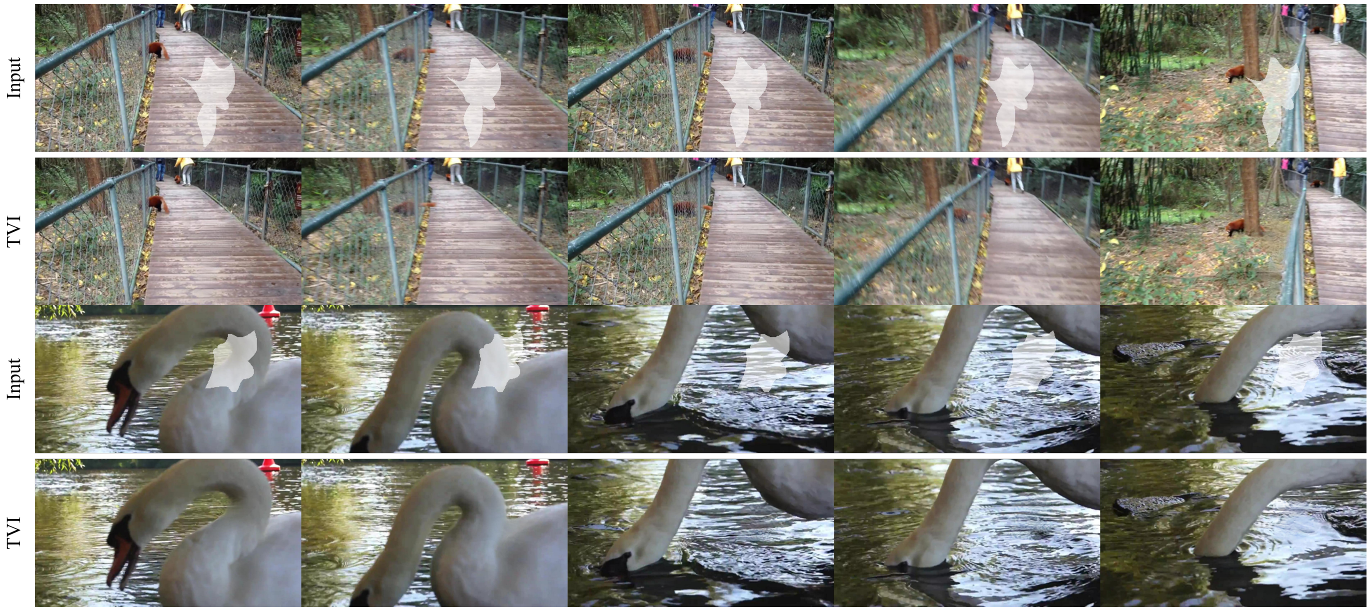

Fig. 13. Additional results on Youtube-VOS dataset for stationary mask type.

methods qualitatively and quantitatively for object-removal, moving curves, and stationary masks.

References

[1] T. Shiratori, Y. Matsushita, X. Tang, and S. B. Kang, "Video completion by motion field transfer," in *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, vol. 1. IEEE, 2006, pp. 411–418.

[2] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum, "Full-frame video stabilization with motion inpainting," *IEEE Transactions on pattern analysis and Machine Intelligence*, vol. 28, no. 7, pp. 1150–1163, 2006.

[3] 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.

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

[5] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, "Deep video inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5792–5801.

[6] 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*, vol. 33, no. 01, 2019, pp. 5232–5239.

[7] S. Lee, S. W. Oh, D. Won, and S. J. Kim, "Copy-and-paste networks for deep video inpainting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4413–4421.

[8] S. W. Oh, S. Lee, J.-Y. Lee, and S. J. Kim, "Onion-peel networks for deep video completion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4403–4412.

[9] 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.

[10] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[11] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[12] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever, "Generative pretraining from pixels," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1691–1703.

[13] 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.

[14] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 873–12 883.

[15] Y. Jiang, S. Chang, and Z. Wang, "Transgan: Two transformers can make one strong gan," *arXiv preprint arXiv:2102.07074*, 2021.

[16] C. Zheng, T.-J. Cham, and J. Cai, "Tfill: Image completion via a transformer-based architecture," *arXiv preprint arXiv:2104.00845*, 2021.

[17] K. Lin, L. Wang, and Z. Liu, "Mesh graphormer," *arXiv preprint arXiv:2104.00272*, 2021.

[18] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[19] Z. Wu, Z. Liu, J. Lin, Y. Lin, and S. Han, "Lite transformer with long-short range attention," *arXiv preprint arXiv:2004.11886*, 2020.

[20] I. Bello, "Lambdanetworks: Modeling long-range interactions without attention," *arXiv preprint arXiv:2102.08602*, 2021.

[21] L. Chi, B. Jiang, and Y. Mu, "Fast fourier convolution," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[22] 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 *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2149–2159.

[23] 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, pp. 417–424.

[24] S. Esedoglu and J. Shen, "Digital inpainting based on the mumford–shah–euler image model," *European Journal of Applied Mathematics*, vol. 13, no. 4, pp. 353–370, 2002.

[25] 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.

[26] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, p. 24, 2009.

[27] A. Newson, A. Almansa, Y. Gousseau, and P. Pérez, "Non-local patch-based image inpainting," *Image Processing On Line*, vol. 7, pp. 373–385, 2017.

[28] J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, "Image completion using planar structure guidance," *ACM Transactions on graphics (TOG)*, vol. 33, no. 4, pp. 1–10, 2014.

[29] 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.

[30] Y. Zeng, J. Fu, H. Chao, and B. Guo, "Learning pyramid-context encoder network for high-quality image inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1486–1494.

[31] M.-c. Sagong, Y.-g. Shin, S.-w. Kim, S. Park, and S.-j. Ko, "Pepsi: Fast image inpainting with parallel decoding network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 360–11 368.

[32] Y.-G. Shin, M.-C. Sagong, Y.-J. Yeo, S.-W. Kim, and S.-J. Ko, "Pepsi++: Fast and lightweight network for image inpainting," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 252–265, 2020.

[33] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2536–2544.

[34] 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.

[35] Y. Ren, X. Yu, R. Zhang, T. H. Li, S. Liu, and G. Li, "Structureflow: Image inpainting via structure-aware appearance flow," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 181–190.

[36] 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, pp. 5505–5514.

[37] J. Li, N. Wang, L. Zhang, B. Du, and D. Tao, "Recurrent feature reasoning for image inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7760–7768.

[38] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on image processing*, vol. 13, no. 9, pp. 1200–1212, 2004.

[39] 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.

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

[41] C. Gao, A. Saraf, J.-B. Huang, and J. Kopf, "Flow-edge guided video completion," in *European Conference on Computer Vision*. Springer, 2020, pp. 713–729.

[42] Y. Zeng, J. Fu, and H. Chao, "Learning joint spatial-temporal transformations for video inpainting," in *European Conference on Computer Vision*. Springer, 2020, pp. 528–543.

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

[44] H. J. Nussbaumer, "The fast fourier transform," in *Fast Fourier Transform and Convolution Algorithms*. Springer, 1981, pp. 80–111.

[45] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[46] 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 (ECCV)*, 2018, pp. 85–100.

[47] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[48] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[49] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[51] Y.-L. Chang, Z. Y. Liu, K.-Y. Lee, and W. Hsu, "Learnable gated temporal shift module for deep video inpainting," *arXiv preprint arXiv:1907.01131*, 2019.

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

[53] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.

[54] X. Zou, L. Yang, D. Liu, and Y. J. Lee, "Progressive temporal feature alignment network for video inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 448–16 457.

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

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

[57] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, "Video-to-video synthesis," *arXiv preprint arXiv:1808.06601*, 2018.

[58] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.