Spatial-Temporal Transformer for Video Snapshot Compressive Imaging

Lishun Wang, Miao Cao, Yong Zhong, and Xin Yuan, Senior Member, IEEE

Abstract—Video snapshot compressive imaging (SCI) captures multiple sequential video frames by a single measurement using the idea of computational imaging. The underlying principle is to modulate high-speed frames through different masks and these modulated frames are summed to a single measurement captured by a low-speed 2D sensor (dubbed optical encoder); following this, algorithms are employed to reconstruct the desired high-speed frames (dubbed software decoder) if needed. In this article, we consider the reconstruction algorithm in video SCI, i.e., recovering a series of video frames from a compressed measurement. Specifically, we propose a Spatial-Temporal transformer (STFormer) to exploit the correlation in both spatial and temporal domains. STFormer network is composed of a token generation block, a video reconstruction block, and these two blocks are connected by a series of STFormer blocks. Each STFormer block consists of a spatial self-attention branch, a temporal self-attention branch and the outputs of these two branches are integrated by a fusion network. Extensive results on both simulated and real data demonstrate the state-of-the-art performance of STFormer. The code and models are publicly available at https://github.com/ucaswangls/STFormer.

Index Terms—Attention, coded aperture compressive temporal imaging (CACTI), compressive sensing, convolutional neural networks, deep learning, snapshot compressive imaging, transformer

1 INTRODUCTION

With recent advances in artificial intelligence, high-quality, high-dimensional data have become one of the bottlenecks for large-scale deep learning models. In other words, capturing more data with multiple dimensions will lead to a dramatic increase in storage and transmission costs. Unlike ordinary cameras which capture RGB images, computational imaging [1], [2] provides a new way to capture high-dimensional data in a memory-efficient manner. It is promising to support the explosion of artificial intelligence using data captured by computational imaging systems. In this paper, we focus on snapshot compressive imaging (SCI) [3], especially video SCI systems, which are capable of capturing high-speed videos using a low-speed camera [4], enjoying the benefits of low memory requirement, low bandwidth for transmission, low cost and low power.

- Lishun Wang and Yong Zhong are with the Chengdu Institute of Computer Application Chinese Academy of Sciences, Chengdu, Sichuan 610041, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China. E-mail: wangling17@mails.ucas.edu.cn, zhongyong1@csis.com.cn.
- Miao Cao and Xin Yuan are with the Research Center for Industries of the Future Research and School of Engineering, Westlake University, Hangzhou 310030, China. E-mail: caomiao, xyuan@westlake.edu.cn.

Manuscript received 17 May 2022; revised 15 October 2022; accepted 26 November 2022. Date of publication 29 November 2022; date of current version 5 June 2023.

The work of Xin Yuan was supported by NSFC under Grant 62271414, in part by Westlake Foundation under Grant 2021B1501-2, in part by the Research Center for Industries of the Future (RCIF) at Westlake University, and in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LR23F010001. (Corresponding author: Xin Yuan.)

Recommended for acceptance by J. Gu.

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2022.3225382, provided by the authors.

Digital Object Identifier no. 10.1109/TPAMI.2022.3225382.

1.1 Video Snapshot Compressive Imaging

Traditional high-speed camera method for capturing high-speed scenes often faces the disadvantages of high hardware cost, high storage requirement and high transmission bandwidth. Bearing these challenges, video SCI system provides an elegant solution. As shown in Fig. 1, video SCI is an integrated hardware plus software system. For the hardware part (encoder), each frame of the original video is encoded by different masks, then a series of encoded frames is integrated (summed) by the (grayscale or color) camera to generate a compressed measurement. In this manner, video SCI can achieve efficient compression during optical domain imaging and improve the efficiency of video storage and transmission. At present, a variety of video SCI systems [4], [6], [7], [8], [9] have been built. The mask is usually generated by a digital micromirror device (DMD) or a spatial light modulator (SLM), and the encoded measurement is usually captured by a charge-coupled device (CCD) or a complementary metal-oxide semiconductor (CMOS) camera. In addition, for color video SCI [5], there is usually a Bayer-filter before the sensor array to capture different color components such as red, green or blue at different pixel points, and ultimately the camera outputs the modulated Bayer measurement. For the software part (decoder), measurement and masks are usually fed into the reconstruction algorithm to recover the desired high-speed video.

1.2 Reconstruction Algorithms for Video SCI

In the decoding stage, the video SCI reconstruction algorithm aims to solve an ill-posed inverse problem. Traditional model-based reconstruction algorithms combines the idea of iterative optimization, such as generalized alternating projection (GAP) [10] and alternating direction method of multipliers (ADMM) [11], with a variety of prior
knowledge, such as total variation (TV) [12], non-local low rank [13] and Gaussian mixture model [14]. Although these methods do not require any data to train the model, they usually suffer from poor reconstruction quality and long reconstruction time. In recent years, due to the development of deep learning and its strong generalization ability, researchers have constructed many learning-based models. For example, BIRNAT [15] uses convolutional neural networks (CNNs) and bidirectional recurrent neural networks (RNNs) for frame-by-frame reconstruction, while U-net [8] uses a simple U-shaped structure for fast reconstruction. RevSCI [16] can save training memory by using reversible mechanism [17]. Plug-and-play (PnP) methods, such as PnP-FFDNet [18] and PnP-FastDvDnet [19], make the model more flexible by integrating the deep denoising model into an iterative optimization algorithm. GAP-net [20], DUN-3DUnet [21] and Two-stage [22] further improve reconstruction quality and speed by using the ideas of deep unfolding [23]. Although previous methods have made great strides in video SCI reconstruction, at present, video SCI reconstruction still suffers from several key challenges:

1) In the video SCI encoding process, the entire temporal domain information will be compressed, which will bring the effect of spatial-temporal confusion. However, the pseudo-inverse method [10], used in the first step of the video SCI reconstruction process, cannot eliminate the effect of spatial-temporal confusion.

2) Most existing end-to-end deep learning methods can not adapt to masks and input sizes changes. More specifically, the model usually needs to be retrained or fine-tuned when masks and input sizes change [24].

3) PnP methods usually suffer from excessive smoothing and loss of local details.

1.3 Motivation of Using Transformer

In the video SCI encoding process, the entire temporal domain information will be compressed. This will bring the effect of spatial-temporal confusion, which is not limited to the local adjacent frames, but comes from the entire temporal domain. However, the pseudo-inverse method, used in the first step of the video SCI reconstruction, cannot eliminate the effect of spatial-temporal confusion. It is well known that the non-local similarity plays a significant role in eliminating the effect of spatial-temporal confusion [13]. Although convolution can exploit the non-local information to some extent, it is limited by the receptive field. By contrast, Transformer [25] can be used to exploit the long-range non-local similarity, and hence can improve the reconstruction results significantly. More importantly, in video SCI reconstruction, we not only need to exploit the spatial non-local information, but also the temporal non-local information.

Bearing these concerns in mind, we utilize Transformer to build an efficient video SCI reconstruction network due to its powerful model capacity and strong ability to explore long-term temporal dependencies.

1.4 Contributions of This Paper

In this paper, we propose a simple yet effective network for video SCI reconstruction. Our main contributions are as follows:

- We build an end-to-end deep video SCI reconstruction network based on spatial-temporal Transformer, which can be well applied to multiple video SCI reconstruction tasks.
- Through space-time factorization and local self-attention mechanism, we propose an efficient and flexible Transformer module, dubbed STFormer, with linear complexity, which can explore spatial-temporal correlations efficiently.
- We propose a Grouping Resnet Feed Forward module to further improve reconstruction quality by fusing multi-layer information and strengthening the correlation between different layers.
- We use multiple 3D convolutions to construct a token generation block to prevent loss of local details.
- Experimental results on a large number of simulated and real datasets demonstrate that our proposed algorithm achieves state-of-the-art (SOTA) results with better real-time performance compared to previous methods.

The rest of this paper is organized as follows: Section 2 introduces the mathematical model of grayscale and color video SCI and the related work of vision Transformers. In Section 3, we propose a reconstruction model STFormer for different video SCI reconstruction tasks, including grayscale video, color video and large-scale video. Section 4 shows experimental results on multiple simulated and real datasets. Section 5 concludes the entire paper.

2 RELATED WORK

Since video SCI is an integrated system of hardware and software, in the following, we first introduce the mathematical model of video SCI, and then briefly review the related work of vision Transformer, which is the main part of the reconstruction network.

2.1 Review of Mathematical Model for Video SCI

Video SCI system encodes high-dimensional video data into a 2D measurement, and coded aperture compressive temporal imaging (CACTI) [4] is one of the earliest video SCI systems. As shown in Fig. 1, the three-dimensional video data is first modulated by multiple masks. Then, the encoded high-resolution image is sampled in the spatial domain. However, the pseudo-inverse method, used in the first step of the video SCI reconstruction, cannot eliminate the effect of spatial-temporal confusion. It is well known that the non-local similarity plays a significant role in eliminating the effect of spatial-temporal confusion [13]. Although convolution can exploit the non-local information to some extent, it is limited by the receptive field. By contrast, Transformer [25] can be used to exploit the long-range non-local similarity, and hence can improve the reconstruction results significantly. More importantly, in video SCI reconstruction, we not only need to exploit the spatial non-local information, but also the temporal non-local information.

Bearing these concerns in mind, we utilize Transformer to build an efficient video SCI reconstruction network due to its powerful model capacity and strong ability to explore long-term temporal dependencies.

1.4 Contributions of This Paper

In this paper, we propose a simple yet effective network for video SCI reconstruction. Our main contributions are as follows:

- We build an end-to-end deep video SCI reconstruction network based on spatial-temporal Transformer, which can be well applied to multiple video SCI reconstruction tasks.
- Through space-time factorization and local self-attention mechanism, we propose an efficient and flexible Transformer module, dubbed STFormer, with linear complexity, which can explore spatial-temporal correlations efficiently.
- We propose a Grouping Resnet Feed Forward module to further improve reconstruction quality by fusing multi-layer information and strengthening the correlation between different layers.
- We use multiple 3D convolutions to construct a token generation block to prevent loss of local details.
- Experimental results on a large number of simulated and real datasets demonstrate that our proposed algorithm achieves state-of-the-art (SOTA) results with better real-time performance compared to previous methods.

The rest of this paper is organized as follows: Section 2 introduces the mathematical model of grayscale and color video SCI and the related work of vision Transformers. In Section 3, we propose a reconstruction model STFormer for different video SCI reconstruction tasks, including grayscale video, color video and large-scale video. Section 4 shows experimental results on multiple simulated and real datasets. Section 5 concludes the entire paper.
speed scene is captured by a two-dimensional camera sensor through integration across the time dimension.

For grayscale video SCI system, let \( X \in \mathbb{R}^{n_x \times n_y \times B} \) denote the \( B \)-frame (grayscale) video data to be captured, \( M \in \mathbb{R}^{n_x \times n_y \times B} \) denote pre-defined masks. For each frame within the video \( f = 1, \ldots, B \), it is modulated by a mask in the image plane conducted by a 4-f system, and we can express this modulation as

\[
X'(\cdot, f) = X(\cdot, f) \odot M(\cdot, f),
\]

where \( X' \in \mathbb{R}^{n_x \times n_y \times B} \) denotes the modulated video data, \( X(\cdot, f) \) denotes the \( f \)-th frame of the 3D video data \( X, M(\cdot, f) \) denotes the \( f \)-th mask of the 3D mask \( M \), and \( \odot \) denotes the element-wise multiplication.

After this, by compressing across the time domain, the camera sensor plane captures a 2D compressed measurement \( Y \in \mathbb{R}^{n_x \times n_y} \), which can be expressed as

\[
Y = \sum_{f=1}^{B} X'(\cdot, f) + Z,
\]

where \( Z \in \mathbb{R}^{n_x \times n_y} \) denotes the measurement noise.

To keep the notations simple, we further give the vectorized formulation expression of Eq. (2). First, we define \( \text{vec}(\cdot) \) as a vectorization operation of ensued matrix. Then we vectorize

\[
y = \text{vec}(Y) \in \mathbb{R}^{n_x \times n_y},
\]

\[
z = \text{vec}(Z) \in \mathbb{R}^{n_x \times n_y},
\]

\[
x = [x_1, \ldots, x_B]^T \in \mathbb{R}^{n_x \times n_y B},
\]

where \( x_f = \text{vec}(X(\cdot, f)) \). In addition, we define sensing matrix generated by masks in SCI system as

\[
H = [D_1, \ldots, D_B] \in \mathbb{R}^{n_x \times n_y B},
\]

where \( D_f = \text{Diag}(\text{vec}(M_f)) \in \mathbb{R}^{n_x \times n_y B} \) is a diagonal matrix and its diagonal elements is filled by \( \text{vec}(M_f) \). Finally, the vectorization expression of Eq. (2) is

\[
y = Hx + z.
\]

After obtaining the measurement \( y \) (captured by the camera), the next task is to develop a decoding algorithm, i.e., given \( y \) and \( H \), solve \( x \).

For color video SCI system, we use the Bayer pattern filter sensor, where each pixel captures only the red (R), green (G) or blue (B) channel of the raw data in a spatial layout such as ‘RGGB’. Since the colors of adjacent pixels are discontinuous, we divide the original measurement \( Y \) into four sub-measurements \( \{Y^r, Y^g, Y^b, Y^b\} \in \mathbb{R}^{n_x \times n_y} \). Correspondingly, the original masks \( M \) and original video frames \( X \) can also be divided into \( \{M^r, M^g, M^b, M^b\} \in \mathbb{R}^{n_x \times n_y B} \) and \( \{X^r, X^g, X^b, X^b\} \in \mathbb{R}^{n_x \times n_y B} \), respectively. The forward model of each channel can be expressed as

\[
Y^r = \sum_{f=1}^{B} X_f^r \odot M_f^r + Z^r,
\]

\[
Y^g = \sum_{f=1}^{B} X_f^g \odot M_f^g + Z^g,
\]

\[
Y^b = \sum_{f=1}^{B} X_f^b \odot M_f^b + Z^b.
\]

For reconstruction, most previous algorithms [5, 18] reconstruct each sub-measurement separately, and then use off-the-shelf demosaicing algorithms to obtain the desired RGB color videos. This channel-separated reconstruction algorithm cannot make good use of the correlation between channels and is inefficient. Therefore, we input all sub-measurements into our proposed reconstruction network simultaneously, and the reconstruction network directly outputs the desired RGB color video without employing previous demosaicing algorithms.

### 2.2 Vision Transformers

Compared with convolutional networks [26, 27, 28], Vision Transformer (ViT) [29] and its variants [30, 31, 32, 33, 34] have achieved competitive results in the field of image classification [35]. However, the original ViT model training requires a large dataset (e.g., JFT-300M [36]) to achieve good results, DeiT [37] introduced some training strategies to enable the ViT model to achieve better performance in the ImageNet-1K dataset [35], but it still used the global self-attention mechanism, and the computational complexity increases quadratically with the image size. These greatly limit the application of ViT to dense prediction tasks such as object detection [38], semantic segmentation [39, 40, 41], and low-level tasks such as image denoising [42] and image restoration [43, 44].

Moreover, PVT V2 [45] proposed a linear spatial reduction attention (SRA) layer, which uses average pooling to reduce spatial dimension, and achieve improvements on fundamental vision tasks such as classification, detection, and segmentation. MST [46] proposed a spectral-wise multi-head self-attention (S-MSA) and customized a mask-guided mechanism (MM), which effectively explores the application of Transformer to the spectral SCI task [47]. Swin Transformer [48] limits the self-attention calculation to the local window through the division of non-overlapping local windows and shifted window mechanism, which reduces the computational complexity of the Transformer to a linear relationship with the image size. However, Swin Transformer can only explore spatial correlation. Video Swin Transformer [49] can obtain local-scale temporal correlation by extending the local window to the temporal domain, but still cannot achieve good results in long-term correlation scenarios. TimeSformer [50] investigates the spatial-temporal correlation between tokens through a space-time factorization mechanism, and has achieved extremely competitive results in the field of video recognition. However, due to the use of global self-attention mechanism in the spatial domain, the computational complexity becomes too high. Therefore, we need to explore a new vision transformer model that can better exploit both spatial and temporal information.
Complexity grows quadratically with the image size, and its positional embedding size is fixed, lacking the flexibility to change with input size. In addition, most of the existing Transformers divide images or videos into non-overlapping patches to generate tokens, resulting in loss of local details and cannot be well applied to video reconstruction tasks in this work.

3 PROPOSED STFORMER NETWORK FOR VIDEO SCI

Inspired by recent advances in Transformers, especially variants for video processing tasks, in this section, we introduce our proposed spatial-temporal Transformer in detail, which can efficiently explore long-range spatial-temporal correlations and lead to SOTA results on video SCI tasks.

3.1 Overall Architecture

The overall video reconstruction network architecture is depicted in Fig. 2. In the preprocessing stage shown in Fig. 2a, we design the input initialization block with reference to GAP-net and Two-stage. By pre-processing masks (M) and measurement (Y) with the input initialization block, we can get a coarse estimation of the modulated frames $\hat{X}$, where $IC = 1$ represents the input channel number. Then, $\hat{X}$ is fed into the token generation block.
SoftMax \( H = \frac{1}{\text{softmax}(H)} \)

The value \((18)\) represents the learnable relative position encoding.

3.2 StFormer Block

As shown in Fig. 2d, the StFormer block is composed of three parts: spatial self-attention (SSA) branch, temporal self-attention (TSA) branch, and grouping resnet feed forward (GRFF) network. Among them, the SSA block can well explore the spatial correlation between tokens, and TSA block can well establish the temporal correlation between tokens, respectively. GRFF network can further investigate the correlation between adjacent tokens.

### 3.2.1 Spatial Self-Attention Branch

Most of previous Transformer architectures used a global self-attention mechanism to calculate the correlation between tokens, which will cause the computational complexity of the Transformer model to increase quadratically with the number of tokens. This limits the application of Transformer to video SCI reconstruction tasks.

Bearing this in mind, in our proposed spatial self-attention branch, we use a local self-attention method to calculate the spatial correlation. As shown in Fig. 2d, in SSA branch, we first divide the feature map \( X_s \in \mathbb{R}^{B \times H \times W \times C} \) into a series of non-overlapping local windows \( X_s \in \mathbb{R}^{L \times J \times S \times C} \), where \( J = G_h \times G_w \times G_d \) represents the number of tokens in each local window, \( L = B / J \) represents the number of local windows, and \( G_h, G_w, G_d \) represent the height, width, and timing length of the local window, respectively. Since spatial self-attention only calculates the spatial correlation between tokens, we set \( G_d \) to 1, and the default values of \( G_h \) and \( G_w \) are 7. Then, the self-attention computation is restricted to each non-overlapping local window.

For the self-attention calculation of the local window, we first linearly map \( X_s \) to get query \( Q_s \in \mathbb{R}^{L \times J \times S \times C} \), key \( K_s \in \mathbb{R}^{L \times J \times S \times C} \) and value \( V_s \in \mathbb{R}^{L \times J \times S \times C} \):

\[
Q_s = X_s W^Q_s, \\
K_s = X_s W^K_s, \\
V_s = X_s W^V_s, 
\]

where \( W^Q_s \in \mathbb{R}^{C \times C}, W^K_s \in \mathbb{R}^{C \times C} \) and \( W^V_s \in \mathbb{R}^{C \times C} \) represent projection matrices and share parameters between different windows. Then, we respectively divide \( Q_s, K_s, V_s \) into \( N \) heads along the feature channel \( C \):

\[
Q_s = [Q^{1}_s, \ldots, Q^{N}_s], \\
K_s = [K^{1}_s, \ldots, K^{N}_s], \\
V_s = [V^{1}_s, \ldots, V^{N}_s],
\]

and the feature dimension of each head becomes \( d_h = \frac{C}{N} \). For each head \( i = 1, \ldots, N \), the attention can be calculated by the local self-attention mechanism as:

\[
\text{Attention}(Q_i, K_i, V_i) = \text{SoftMax}(Q_i^T K_i \sqrt{d} + b^r_i) V_i,
\]

where \( K_i^T \) represents the transpose of matrix \( K_i \) and \( b^r_i \in \mathbb{R}^{J \times J} \) represents the learnable relative position encoding. After this, we concatenate the outputs of \( N \) heads along the channel dimension and conduct a linear mapping to obtain the final output \( X_s \in \mathbb{R}^{B \times H \times W \times C} \) of spatial local window multi-head self-attention mechanism (SLW-MSA):

\[
X_s = W^R(\text{Concat}(H^1_s, \ldots, H^N_s)),
\]

where

**Note:** The equations and explanations given above are based on the understanding of the document content. The specific implementation details and mathematical expressions may vary based on the actual context and application requirements.
\[
X_s = \text{SLW-MSA}(X_s).
\] (20)

For the lack of connection between local windows, we refer to the shifted window partitioning approach of Swin Transformer [49] to establish the information interaction between local windows.

### 3.2.2 Temporal Self-Attention Branch

Previous reconstruction algorithms typically use 2D or 3D convolutions to explore temporal correlations. Due to the local connection of convolutions, its receptive field is limited, which makes it incapable of investigating long-term correlation. Therefore, we exploit the long-term perception ability of Transformer to build a temporal self-attention branch. Different from the SSA branch, the TSA branch only performs self-attention on tokens in the same spatial position. In other words, the TSA branch does not calculate the spatial correlation between tokens.

As shown in Fig. 2d, in TSA branch, for temporal self-attention calculation, we reshape the input \( X_t \) to \( X_t \in \mathbb{R}^{HW \times D \times C} \). After this, similar to the local window self-attention calculation of the SSA branch, we first linearly map \( X_t \) to get \( \text{query } Q_t \in \mathbb{R}^{HW \times D \times C} \), \( \text{key } K_t \in \mathbb{R}^{HW \times D \times C} \), and \( \text{value } V_t \in \mathbb{R}^{HW \times D \times C} \).

\[
Q_t = X_t W_t^Q, \quad K_t = X_t W_t^K, \quad V_t = X_t W_t^V,
\] (21)

where \( W_t^Q \in \mathbb{R}^{C \times C}, W_t^K \in \mathbb{R}^{C \times C} \) and \( W_t^V \in \mathbb{R}^{C \times C} \) represent projection matrices and share parameters between different temporal windows.

Different from the SSA branch, we decrease the channel dimensions of \( Q_t, K_t \) and \( V_t \) to further reduce the computational complexity. Then, we respectively divide \( Q_t, K_t, V_t \) into \( N \) heads along the feature channel: \( Q_t = [Q_t^{(1)}, \ldots, Q_t^{(N)}], \quad K_t = [K_t^{(1)}, \ldots, K_t^{(N)}], \quad V_t = [V_t^{(1)}, \ldots, V_t^{(N)}] \), and the feature dimension of each head becomes \( d_h = \frac{C}{N} \). For each head \( j = 1, \ldots, N \), the attention can be calculated by the temporal window self-attention mechanism as:

\[
\text{Attention}(Q_t^{(j)}, K_t^{(j)}, V_t^{(j)}) = \text{SoftMax}(Q_t^{(j)} K_t^{(j)T} / \sqrt{d} + B_t^{(j)}) V_t^{(j)},
\] (22)

where \( B_t^{(j)} \in \mathbb{R}^{D \times D} \) represents the learnable relative position encoding. Then, we concatenate the outputs of \( N \) heads along the channel dimension and perform a linear mapping to obtain the final output \( X_t \in \mathbb{R}^{D \times H \times W \times C} \) of temporal window multi-head self-attention mechanism (TW-MSA):

\[
\hat{X}_t = \mathbf{R}(W_t^P (\text{Concat}[H_t^{(1)}, \ldots, H_t^{(N)}]))
\] (23)

where \( H_t^{(j)} = \text{Attention}(Q_t^{(j)}, K_t^{(j)}, V_t^{(j)}), \quad W_t^P \in \mathbb{R}^{C \times C} \) represents projection matrices, and \( \mathbf{R} \) represents reshape operation. The whole process of the temporal self-attention branch can be expressed as:

\[
X_t = \text{TW-MSA}(X_t).
\] (24)

### 3.2.3 Grouping Resnet Feed Forward Network

The feed forward (FF) network of a regular Transformer [25] uses two linear mapping layers to transform features; the first linear layer expands the channel dimension (usually by a factor of 4), and the second linear layer restores the channel dimension to the original one [44]. In the whole FF network process, the operation of each feature point are independent of each other, and there will be no interaction between feature points. In this work, we modify the original FF network with details shown in the GRFF network in Fig. 2d. We divide the input feature map into two parts along the channel dimension; the first part of the feature is sent to a Resnet module, whose output is added to the second part of the feature, and the summed feature is used as input to another Resnet module. Following this, the outputs of the two Resnet modules are concatenated along the channel dimension to obtain the final output of the GRFF network. Given an input \( X_g \in \mathbb{R}^{D \times H \times W \times C} \), the whole GRFF network process can be expressed as:

\[
X_{s1}, X_{s2} = \text{Split}(X_g),
\] (25)

\[
Y_{r1} = \text{Resnet}_1(X_{s1}),
\] (26)

\[
Y_{r2} = \text{Resnet}_2(X_{s2} + Y_{r1}),
\] (27)

\[
X_g = \text{Concat}([Y_{r1}, Y_{r2}]),
\] (28)

where \( \text{Split} \) represents feature division along the channel, \( \text{Concat} \) represents feature merging across the channel dimension and \( X_g \in \mathbb{R}^{D \times H \times W \times C} \) represents the output of the GRFF network. Overall, the GRFF network conducts more feature mapping by grouping and using the Resnet mechanism to achieve multi-layer information fusion, and the use of convolution enhances the information interaction between adjacent feature points. All of these are beneficial to improve the quality of video SCI reconstruction.

### 3.2.4 Whole Process of STFormer Block

In summary, the entire process of the STFormer block can be expressed as:

\[
X_{in} = \text{Norm}(X_{in}),
\] (29)

\[
X_s = \text{SLW-MSA}(X_s),
\] (30)

\[
X_t = \text{TW-MSA}(X_t),
\] (31)

\[
X_{st} = X_{in} + (X_s + X_t),
\] (32)

\[
\hat{X}_g = \text{GRFF}(\text{Norm}(X_{st})),
\] (33)

\[
X_{out} = X_{st} + \hat{X}_g,
\] (34)

where \( \text{GRFF}(\cdot) \) represents the GRFF network processing, specifically described in Eqs. (25)–(28). \( \text{Norm}(\cdot) \) represents the Layer Normalization [55], and \( X_{out} \) is the output of STFormer block.

### 3.2.5 Computational Complexity

In addition, we analyze the computational complexity of the spatial-temporal multi-head self-attention mechanism (ST-MSA), which consists of SLW-MSA and TW-MSA:

\[
\Omega(\text{SLW-MSA}) = 4HWDC^2 + 2G_w G_w HWDC,
\] (35)

\[
\Omega(\text{TW-MSA}) = 2HWDC^2 + HWDC^2 C,
\] (36)

\[
\Omega(\text{ST-MSA}) = \Omega(\text{SLW-MSA}) + \Omega(\text{TW-MSA})
\] (37)

\[
= 6HWDC^2 + 2G_w G_w HWDC + HWDC^2 C,
\] (38)
where $\Omega(\cdot)$ represents the computational complexity, $G_h$ and $G_w$ are generally set to 7, and $D$ is the number of frames, and compare it with the global multi-head self-attention mechanism (G-MSA) \[46\],

\[
\Omega(\text{G-MSA}) = 4\text{HWDC}^2 + (\text{HWD})^2C. \quad (39)
\]

We can observe that the computational complexity of our proposed ST-MSA grows linearly with the spatial size $HW$, which is more computationally efficient than G-MSA (quadratic to $HW$).

### 4 Experimental Results

In this section, we compare the performance of the proposed STFormer network with several SOTA video reconstruction methods on multiple simulation and real datasets. The peak-signal-to-noise-ratio (PSNR) and the structured similarity index metrics (SSIM) [56] are used to evaluate the performance of different video SCI reconstruction methods on simulation datasets.

#### 4.1 Datasets

We use DAVIS2017 [57] as the training dataset for the model, which contains 90 different scenes with two resolutions: $480 \times 894$ and $1080 \times 1920$. For the grayscale simulation video testing datasets, we used six benchmark datasets including Kobe, Runner, Drop, Traffic, Aerial and Vehicle with a size of $256 \times 256 \times 8$, following the setup in [18]. For the color simulation video testing datasets, we follow PnP-FastDVNet [19], using six benchmark color simulation datasets, including Beauty, Bosphorus,Jockey, Runner, ShakeNDry and Traffic with a size of $512 \times 512 \times 3 \times 8$. For the large-scale simulation video testing datasets, we used 4 benchmark large-scale simulation datasets, including Messi, Hummingbird, Swinger, Football used in [19].

For the real datasets, we used four datasets, including Duomino, Water Balloon, Hand and Hammer, captured by real video SCI cameras [5], [8] for grayscale and color video, respectively.
TABLE 3
Reconstruction Quality of Three Different Transformers on 6 Grayscale Benchmark Datasets Showing Average PSNR in dB and SSIM

| Dataset | TimeSformer [50] | Video Swin Transformer [49] | STFormer-B |
|---------|------------------|-----------------------------|------------|
| Kobe    | 31.12, 0.932     | 30.72, 0.924                | 35.53, 0.973 |
| Traffic | 27.50, 0.917     | 27.26, 0.911                | 32.15, 0.967 |
| Runner  | 37.13, 0.974     | 37.25, 0.971                | 42.64, 0.988 |
| Drop    | 40.14, 0.988     | 39.82, 0.987                | 45.08, 0.995 |
| Crash   | 28.11, 0.931     | 28.46, 0.939                | 31.06, 0.970 |
| Aerial  | 28.96, 0.915     | 29.07, 0.915                | 31.56, 0.953 |
| Average | 32.16, 0.943     | 32.09, 0.941                | 36.34, 0.974 |

4.2 Implementation Details

During training, we perform data augmentation on DAVIS2017 using random horizontal flipping, random scaling, and random cropping. Following the CACTI imaging process, a series of measurements are generated. We use measurement masks as inputs to train the STFormer network and use Adam optimizer [58] to optimize the model. Since our proposed STFormer network is flexible in input size, to speed up model training, we first train the model on data with a spatial resolution of 128 x 128 (100 epochs with an initial learning rate set be 0.0001) and then fine-tune it on data with a spatial resolution of 256 x 256 (20 epochs with an initial learning rate set be 0.00001). All experiments are run on PyTorch framework with 8 NVIDIA RTX 3090 GPUs.

4.3 Results on Simulation Datasets

In this section we present the results from the simulation to real dataset, first in grayscale and then in color.

To trade-off speed and performance, we have trained three models with different size, dubbed as STFormer-L, STFormer-B and STFormer-S, standing for Large, Base and Small networks respectively. The hyper-parameters of these models are as follows:

- STFormer-S: $C = 64$, block numbers = [2,2,2,2],
- STFormer-B: $C = 256$, block numbers = [2,2,2,2],
- STFormer-L: $C = 256$, block numbers = [4,4,4,4],

where $C$ represents the number of input channels of the STFormer block. The model parameters (Params) and theoretical computational complexity (FLOPs) are shown in Table 2, where we can observe that the parameters and computation of our proposed STFormer-S network are less than BIRNAT and RevSCI, and the STFormer-B network is less than DUN-3DUnet.

4.3.1 Grayscale Simulation Video

Currently, there are various methods for video SCI reconstruction, here we compare our method with some SOTA methods, e.g., model-based iterative optimization methods (GAP-TV [12] and DeSCI [13]), end-to-end deep learning methods (U-net [8], MetaSCI [24], BIRNAT [15] and RevSCI [16]), plug-and-play methods (PnP-FFDNet [18], PnP-FastDVDNet [19]) and deep unfolding methods (GAP-net [20], DUN-3DUnet [21]). Table 1 presents the average PSNR and SSIM values for different reconstruction methods on 6 benchmark grayscale datasets and the average reconstruction time for a single measurement. Fig. 4 shows the visualization results of several SOTA reconstruction methods. We can observe that our proposed method achieves higher reconstruction quality and better real-time performance than previous SOTA methods by a large margin. From the visualization of reconstructed videos, our proposed method can recover more details and edge information. We summarize the observations in Table 1 and Fig. 4 as follows:

1) Our proposed method (STFormer-L) achieves an average PSNR value of 36.75 dB and an SSIM value of 0.976. Compared with the previous SOTA method DUN-3DUnet (best published results) and the end-to-end deep learning method RevSCI, our proposed method achieves 1.49 dB and 2.83 dB higher average PSNR, respectively.

2) For reconstruction running time, our proposed method achieves a good balance between reconstruction quality and running performance. The reconstruction quality of our proposed STFormer-B model is higher than 36 dB, and the running time is within 500 ms. The running speed and reconstruction quality of the STFormer-S model are higher than most current reconstruction algorithms. Although U-net, MetaSCI, and GAP-net run faster, these algorithms have poor reconstruction quality, with an average PSNR value of less than 33 dB, which is more than 3 dB lower than STFormer-L. In some complex high-speed scenarios, such as Traffic, Crash, Aerial datasets, the reconstruction quality of these methods cannot even reach 29 dB.

3) Benefit from the powerful model capacity of Transformer and its ability to effectively explore long-term dependencies, STFormer has excellent performance in complex scenarios (such as Aerial data), and high-speed scenarios (such as Crash data). The reconstruction quality of these two datasets reaches 31 dB for the first time. From the visualization frames in Fig. 4, we can recover clear edges of tree trunks in Aerial data and the marks on the vehicles in the Crash data. Previous SOTA methods are unable to reconstruct these details, which often leads to excessive smoothing.

To further explore the effectiveness of our proposed STFormer network for video SCI, we directly apply several SOTA video Transformer models, specifically TimeSformer [50] and Video Swin Transformer [49], to video SCI reconstruction tasks. For the original Video Swin Transformer, since the hierarchical structure makes the spatial resolution of the model output too low, we use U-net [59] to upsample the deep features and fuse them with the shallow features. In this way, the model can predict the final reconstruction result. As shown in Table 3, the reconstruction quality of our proposed STFormer is significantly better than that of TimeSformer and Video Swin Transformer, which further verifies the effectiveness of the STFormer network for video SCI.

4.3.2 Color Simulation Video

To verify the effectiveness of our method on various video SCI reconstruction tasks, we extend the STFormer network to the color SCI reconstruction task. We conduct related experiments on six benchmark color RGB datasets [19] with a spatial size of $512 \times 512 \times 3$, where 3 represents the RGB channels. Similar to grayscale video, we compress the video with a compression rate of $B = 8$. As shown in Fig. 1, we

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
capture compressed Bayer measurements using a camera with a Bayer filter. For each dataset with 32 color video frames, we can get 4 Bayer measurements.

Since STFormer is flexibility with respect to input size, in order to speed up training and save memory, we use the approach described in Section 4.2 to train the model on small-scale data. Considering the inflexibility of DUN-3DUnet [21] and RevSCI [16] for input size and masks, training a model with a spatial size of 512 × 512 requires a large amount of memory or training time. We only compare with iterative optimization algorithms (GAP-TV [12], DeSCI [13], PnP-FFDNet-gray [18], PnP-FFDNet-color [18], STFormer-B) on several benchmark color video simulation datasets (Beauty, Bosphorus, Jockey, Runner, ShakeNDry, Traffic). Zoom in for a better view.

### TABLE 4
The Average PSNR in dB (Left Entry), SSIM (Right Entry) and Running Time per Measurement of Different Algorithms on 6 Benchmark Color Datasets

| Dataset       | Beauty       | Bosphorus   | Jockey       | Runner       | ShakeNDry   | Traffic      | Average     | Running time(s) |
|---------------|--------------|-------------|--------------|--------------|-------------|--------------|-------------|-----------------|
| GAP-TV [12]   | 33.08, 0.964 | 29.70, 0.914 | 29.48, 0.887 | 29.10, 0.876 | 29.59, 0.893 | 22.84, 0.643 | 28.47, 0.864 | 10.80 (GPU)     |
| DeSCI [13]    | 34.66, 0.971 | 32.88, 0.952 | 34.14, 0.938 | 36.16, 0.949 | 30.94, 0.905 | 24.62, 0.839 | 32.23, 0.926 | 92640 (GPU)    |
| PnP-FFDNet-gray [18] | 33.21, 0.963 | 28.83, 0.905 | 32.30, 0.918 | 30.83, 0.888 | 27.87, 0.864 | 21.03, 0.711 | 28.93, 0.874 | 13.20 (GPU)    |
| PnP-FFDNet-color [18] | 34.15, 0.967 | 33.06, 0.957 | 34.80, 0.943 | 35.32, 0.940 | 32.37, 0.940 | 24.55, 0.837 | 32.38, 0.931 | 97.80 (GPU)    |
| PnP-FastDvdnet-gray [19] | 33.01, 0.963 | 30.95, 0.954 | 33.31, 0.928 | 32.82, 0.900 | 29.92, 0.892 | 22.81, 0.776 | 30.50, 0.899 | 19.80 (GPU)    |
| PnP-FastDvdnet-color [19] | 33.27, 0.972 | 37.24, 0.976 | 33.63, 0.920 | 33.22, 0.905 | 33.71, 0.967 | 27.49, 0.913 | 34.60, 0.925 | 99.03 (GPU)    |
| BIRNAT-color [60] | 36.08, 0.975 | 38.30, 0.982 | 36.51, 0.956 | 39.65, 0.973 | 34.26, 0.951 | 28.03, 0.915 | 35.47, 0.959 | 0.9 (GPU)      |
| STFormer-S     | 36.83, 0.980 | 38.36, 0.981 | 37.09, 0.963 | 40.56, 0.960 | 34.67, 0.952 | 29.00, 0.923 | 36.09, 0.963 | 0.54 (GPU)     |
| STFormer-B     | 37.37, 0.981 | 40.39, 0.988 | 38.32, 0.968 | 42.45, 0.985 | 35.18, 0.956 | 30.24, 0.939 | 37.32, 0.970 | 1.95 (GPU)     |

Best results are in bold and the second-best results are underlined.
It is worth noting that PnP methods are divided into gray version and color version according to the use of grayscale denoiser or color denoiser. The reconstruction results of different algorithms are shown in Fig. 5 and Table 4, we can summarize the observations as follows:

1) The PSNR value of the STFormer network reaches 37.32 dB, which is 1.85 dB higher than the previous SOTA algorithm BIRNAT-color, especially in the high-speed motion scene Bosphorus, which is improved by 2.09 dB, and it exceeds 30 dB for the first time on the Traffic dataset. This shows that STFormer is also effective in color high-speed scenes.

2) Regarding the running time of algorithm, the reconstruction of each measurement by the DeSCI algorithm is over 24 hours. Although the GAP-TV and PnP algorithms achieve higher real-time performance, they still take more than 10 seconds, while STFormer-B
reconstruction algorithm only needs 1.95 seconds, which is more than 5 times faster than previous PnP methods. Recently, BIRNAT-color takes advantage of the end-to-end model to further improve the running speed of reconstruction. However, our proposed STFormer-S model achieves higher reconstruction quality and faster real-time performance.

3) From the visualization results, our method can recover sharper edges of datasets Beauty, Jockey, Runner and Traffic, and can recover more local details of datasets Bosphorus, ShakenDry. The reconstructed results of GAP-TV, PnP-FFDNet-gray and PnP-FastDVD-gray methods have some artifacts, while the reconstructed results of PnP-FFDNet-color and PnP-FastDVD-color methods have blurred edges and serious loss of details.

### 4.3.3 Large-Scale Simulation Video

Similar to the benchmark color data, we further extend STFormer to large-scale datasets. Following [19], we used 4

| Mask         | BIRNAT | RevSCI | DUN-3DU.net | STFormer-B |
|--------------|--------|--------|-------------|------------|
| Random Mask₁| 23.15, 0.731 | 18.90, 0.531 | 31.59, 0.934 | 36.28, 0.974 |
| Random Mask₂| 23.09, 0.730 | 18.99, 0.537 | 31.62, 0.934 | 36.33, 0.974 |
| Random Mask₃| 23.08, 0.728 | 18.96, 0.528 | 31.80, 0.935 | 36.32, 0.974 |

Fig. 7. The average reconstruction quality (PSNR in dB) of our proposed STFormer-B with different temporal length and spatial local window sizes on 6 grayscale benchmark datasets. Where T2, T4 and T8 indicate that the temporal window lengths are 2, 4 and 8, respectively.

| Table 8 | Average Reconstruction Quality of the Proposed STFormer-B Without the Input Initialization Block on 6 Grayscale Benchmark Datasets |
|---------|--------------------------------------------------------------------------------------------------|
| Network Input | PSNR | SSIM |
| M ⊕ Y | 36.21 | 0.973 |
| Concat([M ⊕ Y, M]) | 36.25 | 0.973 |

| Table 9 | Reconstruction Quality and Running Time (s) on 6 Grayscale Benchmark Datasets Using STFormer With Different Number of Channels and Blocks |
|---------|--------------------------------------------------------------------------------------------------|
| Model   | Channel | Block | PSNR | SSIM | Running time(s) |
| STFormer-S | 64     | 8     | 33.94 | 0.958 | 0.14 |
| STFormer-B | 256    | 8     | 36.34 | 0.974 | 0.49 |
| STFormer-L | 256    | 16    | 36.75 | 0.976 | 0.92 |

| Table 10 | Reconstruction Quality on 6 Grayscale Benchmark Datasets Using Random Masks, the Average PSNR in dB and SSIM are Shown |
|---------|--------------------------------------------------------------------------------------------------|
| Mask         | BIRNAT | RevSCI | DUN-3DU.net | STFormer-B |
| Random Mask₁| 23.15, 0.731 | 18.90, 0.531 | 31.59, 0.934 | 36.28, 0.974 |
| Random Mask₂| 23.09, 0.730 | 18.99, 0.537 | 31.62, 0.934 | 36.33, 0.974 |
| Random Mask₃| 23.08, 0.728 | 18.96, 0.528 | 31.80, 0.935 | 36.32, 0.974 |

Fig. 8. Comparison of reconstruction results of different reconstruction algorithms (GAP-TV [12], PnP-FFDNet [18], DeSCI [13], BIRNAT [15] STFormer-B) on several real datasets (Duomino, Water Balloon) with compression rate $B = 10$. Zoom in for a better view.

Fig. 9. Comparison of reconstruction results of different reconstruction algorithms (PnP-FFDNet [18], PnP-FastDVDNet [19] STFormer-S) on several real datasets (Duomino, Water Balloon) with compression rate $B = 50$. Zoom in for a better view.

Fig. 10. Comparison of reconstruction results of different reconstruction algorithms (GAP-TV [12], DeSCI [13], PnP-FFDNet [18] and STFormer-S) on real color data (Hammer). Zoom in for a better view.
### Table 11
Running Time (Seconds) of Real Data Using Different Algorithms

| Real Data | Pixel resolution | GAP-TV | DeSCI | PnP-FFDNet | PnP-FastDVDnet | STFormer-S |
|-----------|------------------|--------|-------|------------|----------------|------------|
| Hand10    | $512 \times 512 \times 10$ | 37.8   | 2880.0| 19.3       | 29.5           | 1.5        |
| Hand20    | $512 \times 512 \times 20$ | 88.7   | 4320.0| 42.4       | 63.9           | 1.8        |
| Hand30    | $512 \times 512 \times 30$ | 163.0  | 6120.0| 74.7       | 107.7          | 2.2        |
| Hand50    | $512 \times 512 \times 50$ | 303.4  | 12600.0| 144.5     | 203.9          | 2.7        |

Fig. 11. Reconstructed real video data (Duomino) with compression rate $B$ from 10 to 50. Mesh is added to better visualize motion details. The reconstruction algorithm is STFormer-S.
benchmark large-scale datasets, including Messi, Hummingbird, Swinger with spatial size of $1080 \times 1920 \times 3$ and Football with the spatial size of $822 \times 1920 \times 3$, where 3 represents RGB channels. Similar to the color simulation video, we generate measurements with a compression rate of $B = 8$. Note that this is different from the various compression rates used in [19]. We set $B = 8$ since we can use the same model trained for the mid-scale for the reconstruction of these large-scale datasets. This verifies the scalability and flexibility of our model. We can also train the model for large compression rates, only with additional training time and memory.

Due to the fact that network training of BIRNAT [15] and DUN-3DUnet [21] for large-scale datasets requires a lot of memory, while RevSCI [16] requires a longer training time, here we only compare STFormer with GAP-TV [12], PnP-FFDNet [18] and PnP-FastDVDnet [19]. Table 5 and Fig. 6 show the reconstruction results of these algorithms.

Fig. 12. Reconstructed real video data (Hand) with compression rate $B$ from 10 to 50, Mesh is added to better visualize motion details. The reconstruction model is STFormer-S.
As shown in Fig. 6, we can observe that the results of GAP-TV reconstruction are blurry. PnP-FastDVDnet has some artifacts on some datasets, while our STFormer can achieve more realistic results. More importantly, our reconstruction on these datasets can reach more than 31 dB, which proves that video SCI can be applied to real scenes.

As for the running time, since we use a small version of STFormer, it can provide a more efficient runtime. As shown in Table 5, for the reconstruction of a single measurement, our proposed algorithm takes only 4.25 seconds, which is 49 times faster than the previous SOTA algorithm PnP-FastDVDnet.

### 4.4 Ablation Study

To verify the effect of each module in the proposed STFormer network on the overall reconstruction quality, we conducted some ablation experiments on each module. Table 6 shows the effect of each module on the reconstruction quality using the 6 grayscale benchmark datasets, where ✓ indicates that the reconstructed network includes this module, and no TG block indicates that the token generation method in the original Swin Transformer is used. In addition, MLP^2 and MLP^3 represent FF network expansion factors \( \gamma = 4 \) and \( \gamma = 2 \), respectively. We can get the following observations:

1) **Improvements in Token Generation Block:** Tables 6 g and h show that our proposed TG block gain is 1.30 dB higher than its counterpart. Similarly, combining our TG block with other Transformer blocks can also greatly improve the reconstruction quality (see Tables 6 a and b).

2) **Improvements in STFormer Block:** Here, we mainly compare the STFormer block with Video Swin Transformer block. Tables 6 b and e show that our proposed STFormer block can bring an improvement of at least 0.85 dB. In addition, we also verified the TSA branch of the STFormer block. As shown in Tables 6 d and e, the reconstruction quality of the STFormer block with TSA branch can be improved by about 1.85 dB.

3) **Improvements in Grouping Resnet Feed Forward Network:** We compare the GRFF network with the traditional FF network consisting of MLP layers. As shown in Tables 6, the GRFF network can provide an effective gain of more than 1.19 dB.

To further verify the effect of other block and some hyperparameters on the reconstruction quality, we conduct more ablation experiments.

1) **Impact of input initialization block:** Tables 7 and 8 show the reconstruction quality of our proposed method with other input initialization blocks and without input initialization block respectively. Compared with other input initialization blocks and without input initialization block, our proposed input initialization block can obtain a marginal improvement in the reconstruction quality.

2) **Impact of spatial local window size and temporal window length of STFormer block:** In order to verify the influence of the spatial local window size and temporal window length of the STFormer block on the reconstruction quality, we compare and analyze the reconstruction quality under different local window sizes (window sizes are \( 3 \times 3, 5 \times 5, 7 \times 7 \) and \( 9 \times 9 \) respectively) and temporal window lengths (temporal length are \( 2, 4, 8 \) respectively). As shown in Fig. 7, the reconstruction quality grows with the increase of the spatial local window size and the temporal length. After the spatial local window size reaches \( 7 \times 7 \), the reconstruction quality grows slowly. When the temporal length is increased from 4 to 8, we can see a significant improvement.

3) **Impact of width and depth of STFormer network:** To verify the influence of the width and depth of the STFormer network on the reconstruction quality, we designed three models with different number of channels used to adjust the model width and number of blocks used to adjust the model depth. As shown in Table 9, increasing the width and depth of the model is beneficial to the improvement of the reconstruction quality, but it also leads to a longer running time.

### 4.5 Flexibility

Previous end-to-end learning methods usually require retraining models for different masks and different spatial sizes. When dealing with large-scale data, they often require a large amount of memory and training time, which is inefficient. Transformer has a strong model capacity, and it can dynamically adjust the attention map according to different model inputs [43]. Along with our initialization method, the proposed STFormer network is robust to different masks.

To verify this flexibility, we randomly generate three masks that are not used during training. As shown in Table 10, for different masks, the average PSNR value of the STFormer-B network reconstruction results remains within 0.06 dB, which is better than the current SOTA algorithms. By contrast, the previous end-to-end deep learning methods BIRNAT [15] and RevSCI [16] decrease more than 10 dB, and the deep unfolding DUN-3DUnet [21] decreases more than 3 dB.

Due to this flexibility of our proposed STFormer network, coupled with the use of STFormer block local window and relative position bias, the STFormer network trained on small-scale dataset can be used for large-scale datasets. As shown in Tables 4 and 5, these STFormer models are trained on small-scale data (spatial size less than or equal to \( 256 \times 256 \)), and then directly used to reconstruct data with larger spatial size, all achieving SOTA reconstruction results.

### 4.6 Results on Real Video SCI Data

We validate our proposed method on grayscale and color real data. Since the real video SCI imaging system has uncertain noises, it is more challenging to reconstruct real data.

#### 4.6.1 Grayscale Real Video

For the grayscale real data, we use Duomino, Water Balloon and Hand video data captured by [8]. It is worth noting that similar scenes are captured with different compression ratios, \( i.e., B = \{0, 20, 30, 40, 50\} \) and all snapshot measurements spatial size are \( 512 \times 512 \). As shown in Fig. 8, we first compared the reconstruction results with several SOTA
reconstruction algorithms, namely GAP-TV [12], DeSCI [13], PnP-FFDNet [18] and BIRNAT [15] in the scenes of compression rate $B = 10$. By zooming in on the local area, we can observe that our proposed algorithm can recover clear letters and sharp edges in the Dumino data and Water Balloon data, while the reconstruction results of GAP-TV, PnP-FFDNet, DeSCI and BIRNAT algorithms over-smooth these areas with some artifacts.

In addition, our proposed STFormer network can also achieve good reconstruction results at high compression rates, e.g., at $B = 50$, which further verifies the capability of our method to explore long-term temporal dependencies. Although previous reconstruction algorithms can reconstruct high-compression data, their reconstruction results are too smooth and require a long running time (See Table 11); in particular, PnP-FastDVDnet takes 3.4 minutes, DeSCI algorithm reconstruction time is more than 3 hours and our method only needs 2.7 seconds. Figs. 11 and 12 show the reconstruction results of Hand and Dumino with $B = 10, 20, 30, 50$, respectively. We can observe that our proposed method can well reconstruct the desired high-speed video frames with compression rates from 10 to 50. Compare with previous SOTA method PnP-FastDVDnet, our results can provide clear details of Dumino and the Water Balloon even at $B = 50$. Please refer to Fig. 9.
We have further verified our proposed algorithms on the new video SCI system built at Westlake University similar to [8] but with different masks and different compression rates. Please refer to the reconstructed videos in the supplemental material, available online.

4.6.2 Color Real Video

For the color real data, we use Hammer video data captured by [5]. The spatial resolution of a single Bayer mosaic measurement is $512 \times 512$ and the compression rate $B$ is 22. Since most reconstruction algorithms cannot be applied to the color data, we only compare our method with GAP-TV [12], DeSCI [13] and PnP-FFDNet [18]. Fig. 10 shows the reconstruction results of these algorithms. By zooming in on the local areas, we can see that the reconstruction results of GAP-TV, DeSCI, and PnP-FFDNet methods have some artifacts and blurred edges, but our proposed STFormer method can restore these sharper edges.

Furthermore, to fill the gap of lacking real color data for video SCI, we have built a new video SCI system at Westlake University using RGB sensors but with different masks and different compression rates. The reconstructed videos are shown in Figs. 13 and 14. Comparing with previous SOTA algorithms, the reconstruction results of our proposed STFormer network are closer to the real color and can recover more details. Please refer to the enlarged areas in Figs. 13 and 14.

5 Conclusion

In this article, we present STFormer, a spatial-temporal Transformer, to conduct the reconstruction task of video snapshot compressive imaging. Our proposed STFormer network consists of token generation block, video reconstruction block and a series of STFormer blocks. In particular, each STFormer block restricts the self-attention calculation to the spatial local window and time domain through the space-time factorization and local self-attention mechanism, which improves the computational efficiency and increases the flexibility of the model for multi-scale input. Since STFormer can effectively explore spatial-temporal correlations, it achieves SOTA results on multiple video SCI reconstruction tasks. Especially for complex and high-speed motion scenes, STFormer can achieve a reconstruction quality of more than 30 dB, far exceeding the previous SOTA reconstruction algorithms. Furthermore, STFormer is the first end-to-end deep learning network with flexibility of masks and input scale, while enjoying fast inference, greatly facilitating applications of video SCI system in our daily life.

Although STFormer has achieved satisfactory results on video SCI, the current video SCI reconstruction still faces many difficulties, such as the research of deep learning models suitable for different compression rates. In addition, for the real color data, due to the existence of complicated noise, there is still a huge gap between the reconstruction results and the real scene.

Regarding future work plans, one is to extend our STFormer network to other SCI reconstruction tasks, such as spectral SCI [61], [62], [63], [64], and the other is to use STFormer as a backbone for video action recognition [65], video object tracking [66] and other tasks [67], [68], [69].

References

[1] J. N. Mait, G. W. Euliss, and R. A. Athale, “Computational imaging,” Adv. Opt. Photon., vol. 10, no. 2, pp. 409–483, Jun. 2018.
[2] Y. Altmann, S. McLaughlin, M. J. Padgett, V. K. Goyal, O. A. Hero, and D. Faccio, “Quantum-inspired computational imaging,” Science, vol. 361, no. 6403, 2018, Art. no. eaat2298.
[3] X. Yuan, D. J. Brady, and A. K. Katsaggelos, “Snapshot compressive imaging: Theory, algorithms, and applications,” IEEE Signal Process. Mag., vol. 38, no. 2, pp. 65–88, Mar. 2021.
[4] P. Llull et al., “Coded aperture compressive temporal imaging,” Opt. Exp., vol. 21, no. 9, pp. 10 526–10 545, 2013.
[5] X. Yuan et al., “Low-cost compressive sensing for color video and depth,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2014, pp. 3318–3325.
[6] Y. Hitomi, J. Gu, M. Gupta, T. Mitsunaga, and S. K. Nayar, “Video from a single coded exposure photograph using a learned over-complete dictionary,” in Proc. Int. Conf. Comput. Vis., 2011, pp. 287–294.
[7] D. Reddy, A. Veeraraghavan, and R. Chellappa, “P2C2: Programmable pixel compressive camera for high speed imaging,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. 2011, pp. 329–336.
[8] M. Qiao, Z. Meng, J. Ma, and X. Yuan, “Deep learning for video compressive sensing,” APL Photon., vol. 5, no. 3, 2020, Art. no. 30801.
[9] Y. Sun, X. Yuan, and S. Pang, “Compressive high-speed stereo imaging,” Opt. Exp., vol. 25, no. 15, pp. 18 182–18 190, 2017.
[10] X. Liao, H. Li, and L. Carin, “Generalized alternating projection for weighted-2,1 minimization with applications to model-based compressive sensing,” SIAM J. Imag. Sci., vol. 7, no. 2, pp. 797–823, 2014.
[11] S. Boyd, N. Parikh, and E. Chu, Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. Boston, MA, USA: Now, 2011.
[12] X. Yuan, “Generalized alternating projection based total variation minimization for compressive sensing,” in Proc. IEEE Int. Conf. Image Process., 2016, pp. 2539–2543.
[13] Y. Liu, X. Yuan, J. Suo, D. J. Brady, and Q. Dai, “Rank minimization for snapshot compressive imaging,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 12, pp. 2990–3006, Dec. 2019.
[14] J. Yang et al., “Compressive sensing by learning a Gaussian mixture model from measurements,” IEEE Trans. Image Process., vol. 24, no. 1, pp. 106–119, Jan. 2015.
[15] Z. Cheng et al., “BIRNAT: Bidirectional recurrent neural networks with adversarial training for video snapshot compressive imaging,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 258–275.
[16] Z. Cheng et al., “Memory-efficient network for large-scale video compressive sensing,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 16 246–16 255.
[17] J. Behrmann, W. Grathwohl, R. T. Chen, D. Duvenaud, and J.-H. Jacobsen, “Invertible residual networks,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 572–581.
[18] X. Yuan, Y. Liu, J. Suo, and Q. Dai, “Plug-and-play algorithms for large-scale snapshot compressive imaging,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 1447–1457.
[19] X. Yuan, Y. Liu, J. Suo, F. Durand, and Q. Dai, “Plug-and-play algorithms for video snapshot compressive imaging,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 10, pp. 7093–7111, Oct. 2022.
[20] Z. Meng, S. Jalali, and X. Yuan, “GAP-net for snapshot compressive imaging,” 2020, arXiv:2012.08364.
[21] Z. Wu, J. Zhang, and C. Mou, “Dense deep unfolding network with 3D-CNN prior for snapshot compressive imaging,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 4892–4901.
[22] S. Zheng, X. Yang, and X. Yuan, “Two-stage is enough: A concise deep unfolding network for flexible video compressive sensing,” 2022, arXiv:2201.05810.
[23] C. Yang, S. Zhang, and X. Yuan, “Ensemble learning priors unfolding for scalable snapshot compressive sensing,” in Proc. Eur. Conf. Comput. Vis., 2022, pp. 600–618.
[24] Z. Wang, H. Zhang, Z. Cheng, B. Chen, and X. Yuan, “MetaSCI: Scalable and adaptive reconstruction for video compressive sensing,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 2083–2092.
[25] A. Vaswani et al., “Attention is all you need,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2017, pp. 5998–6008.
[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2012, pp. 1097–1105.
[27] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

[28] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 4700–4708.

[29] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” in Proc. Int. Conf. Learn. Representations, 2020, pp. 1–9.

[30] X. Dong et al., “CSWin transformer: A general vision transformer backbone with cross-shaped windows,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 12–12–134.

[31] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable DETR: Deformable transformers for end-to-end object detection,” in Proc. Int. Conf. Learn. Representations, 2020, pp. 1–9.

[32] L. Yuan et al., “Tokens-to-token ViT: Training vision transformers from scratch on ImageNet,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 558–567.

[33] W. Wang et al., “Pyramid vision transformer: A versatile backbone for dense prediction without convolutions,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021.

[34] K. Han et al., “A survey on vision transformer,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 1, pp. 87–110, 2023.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2009, pp. 248–255.

[36] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, “Revisiting unreasonable effectiveness of data in ImageNet era,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 843–852.

[37] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention,” in Proc. Int. Conf. Mach. Learn., 2021, pp. 1034–1037.

[38] T.-Y. Lin et al., “Microsoft COCO: Common objects in context,” in Proc. Eur. Conf. Comput. Vis., 2014, p. 7407.

[39] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, “Scene parsing through ADE20K dataset,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2017, pp. 633–641.

[40] B. Zhou et al., “Semantic understanding of scenes through the ADE20K dataset,” Int. J. Comput. Vis., vol. 127, no. 3, pp. 302–321, 2019.

[41] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, “CCNet: Criss-cross attention for semantic segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 603–612.

[42] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising,” IEEE Trans. Image Process., vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[43] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, “SwinIR: Image restoration using swin transformer,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 8333–8344.

[44] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, “Restormer: Efficient transformer for high-resolution image restoration,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 5728–5739.

[45] W. Wang et al., “PVT v2: Improved baselines with pyramid vision transformer,” Comput. Vis. Med., vol. 8, pp. 415–424, 2022.

[46] Y. Cai et al., “Masked autoencoders are scalable vision learners,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 10012–10022.

[47] Z. Liu et al., “Swin transformer: Hierarchical vision transformer using shifted windows,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 10012–10022.

[48] Z. Liu et al., “Video swin transformer,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 3202–3211.

[49] G. Bertasius, H. Wang, and L. Torresani, “Is space-time attention all you need for video understanding?” in Proc. Int. Conf. Mach. Learn., 2021, Art. no. 4.

[50] J. Wang et al., “BEVT: BERT pretraining of video transformers,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 1473–1474.  

[51] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 16 000–16 009.

[52] H. Bao, L. Dong, and F. Wei, “BEIT: BERT pre-training of image transformers,” 2021, arXiv:2106.08254.

[53] B. Xu, N. Wang, T. Chen, and M. Li, “Empirical evaluation of rectified activations in convolutional network,” 2015, arXiv:1505.00853.

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

[55] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, Apr. 2004.

[56] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbelaez, A. Sorkine-Hornung, and L. Van Gool, “The 2017 danny challenge on video object segmentation,” 2017, arXiv:1704.00675.

[57] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980.

[58] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in Proc. Int. Conf. Med. Image Comput. Assist. Interv. 2015, pp. 234–241.

[59] Z. Cheng et al., “Recurrent neural networks for snapshot compressive imaging,” IEEE Trans. Pattern Anal. Mach. Intell., early access, Mar. 24, 2022, doi: 10.1109/TPAMI.2022.3161934.

[60] Z. Meng, J. Ma, and X. Yuan, “End-to-end low cost compressive spectral imaging with spatial-spectral self-attention,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 187–204.

[61] L. Wang, Z. Wu, Y. Zhong, and X. Yuan, “Snapshot spectral compressive imaging processing on CAVs,” in Proc. IEEE Trans. Image Process., vol. 8, pp. 415–424, 2022.

[62] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” 2014, arXiv:1502.05843.

[63] B. Xu, N. Wang, T. Chen, and M. Li, “Empirical evaluation of rectified activations in convolutional networks,” 2015, arXiv:1505.00853.

[64] Z. Liu et al., “Video swin transformer,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 4059–4069.

[65] C.-Y. Wu, M. Zanger, H. Hu, R. Mannathu, A. J. Smola, and P. Krähenbühl, “Compressed video action recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 6026–6035.

[66] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, “FairMOT: On the fairness of detection and re-identification in multiple object tracking,” Int. J. Comput. Vis., vol. 129, no. 11, pp. 3069–3087, 2021.

[67] S. Li, X. Yuan, and W. Shi, “An integrated framework for compressive imaging processing on CAVs,” in Proc. ACM/IEEE Symp. Edge Comput., 2020, pp. 125–138.

[68] Z. Zhang et al., “From compressive sampling to compressive tasking: Retrieving semantics in compressed domain with low bandwidth,” Photonics, vol. 3, no. 1, pp. 1–22, 2022.

[69] Z. Chen, S. Zheng, Z. Tong, and X. Yuan, “Physics-driven deep learning enables temporal compressive coherent diffraction imaging,” Optica, vol. 9, no. 6, pp. 677–680, 2022.

Lishun Wang received the BS degree from the Sichuan Agricultural University, in 2017. He is currently working toward the PhD degree with the Chengdu Institute of Computer Application Chinese Academy of Sciences. His research interests include computer vision and computational imaging.

Miao Cao received the BS degree from the China University of Geosciences, Beijing, in 2012. He is currently working toward the PhD degree with Zhejiang University and Westlake University. His research interests include computer vision and computational imaging.
Yong Zhong received the BS and MS degrees from the Chengdu University of Technology, Chengdu, China, in 1989 and 1994, respectively, and the PhD degree from the University of Chinese Academy of Sciences, in 2003. He is currently a professor with the University of Chinese Academy of Sciences. His research interests include deep learning for image processing.

Xin Yuan (Senior Member, IEEE) received the BEng and MEng degrees from Xidian University, in 2007 and 2009, respectively, and the PhD from the Hong Kong Polytechnic University, in 2012. He is currently an associate professor with Westlake University. He was a video analysis and coding lead researcher with Bell Labs, Murray Hill, NJ, USA from 2015 to 2021. Prior to this, he was a post-doctoral associate with the Department of Electrical and Computer Engineering, Duke University from 2012 to 2015. His research interests include computational imaging and machine learning. He has been the associate editor of the *Pattern Recognition* (2019-), *International Journal of Pattern Recognition and Artificial Intelligence* (2020-) and *Chinese Optics Letters* (2021-). He led the special issue of “Deep Learning for High Dimensional Sensing” in the *IEEE Journal of Selected Topics in Signal Processing* in 2022.

For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/csdli](http://www.computer.org/csdli).