BIRNAT: Bidirectional Recurrent Neural Networks with Adversarial Training for Video Snapshot Compressive Imaging

Ziheng Cheng[1][0000−0002−7504−197X], Ruiying Lu[1][0000−0002−8825−6064], Zhengjue Wang[1][0000−0002−1846−495X], Hao Zhang[1][0000−0002−2928−2692], Bo Chen[1⋆][0000−0001−5151−9388], Ziyi Meng[2,3][0000−0001−8294−8847], and Xin Yuan[4⋆][0000−0002−8311−7524]

[1] National Laboratory of Radar Signal Processing, Xidian University, Xian, China
[2] Beijing University of Posts and Telecommunications, Beijing, China
[3] New Jersey Institute of Technology, NJ, USA [4] Nokia Bell Labs, NJ, USA
zhcheng@stu.xidian.edu.cn,
{ruiyinglu_xidian,zhengjuewang,zhanghao_xidian}@163.com,
bchen@mail.xidian.edu.cn,mengziyi@bupt.edu.cn,xyuan@bell-labs.com

Abstract. We consider the problem of video snapshot compressive imaging (SCI), where multiple high-speed frames are coded by different masks and then summed to a single measurement. This measurement and the modulation masks are fed into our Recurrent Neural Network (RNN) to reconstruct the desired high-speed frames. Our end-to-end sampling and reconstruction system is dubbed **BI**directional **R**ecurrent **N**eural networks with **A**dversarial **T**raining (BIRNAT). To our best knowledge, this is the first time that recurrent networks are employed to SCI problem. Our proposed BIRNAT outperforms other deep learning based algorithms and the state-of-the-art optimization based algorithm, DeSCI, through exploiting the underlying correlation of sequential video frames. BIRNAT employs a deep convolutional neural network with Resblock and feature map self-attention to reconstruct the first frame, based on which bidirectional RNN is utilized to reconstruct the following frames in a sequential manner. To improve the quality of the reconstructed video, BIRNAT is further equipped with the adversarial training besides the mean square error loss. Extensive results on both simulation and real data (from two SCI cameras) demonstrate the superior performance of our BIRNAT system. The codes are available at https://github.com/BoChenGroup/BIRNAT.

Keywords: Snapshot compressive imaging, compressive sensing, deep learning, convolutional neural networks, recurrent neural network.

1 Introduction

Videos are essentially sequential images (frames). Due to the high redundancy in these frames, a video codec [25] can achieve a high (> 100) compression rate

⋆ corresponding author.

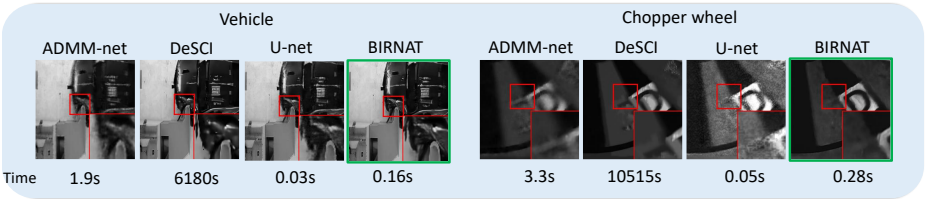

Fig. 1. Selected reconstructed frames using state-of-the-art methods, where DeSCI is an optimization algorithm, ADMM-net and U-net are based on CNNs and BIRNAT (ours) is based on RNNs. Left: simulation data `Vehicle`, the evaluation metric PSNR (in dB) is 23.62 (ADMM-net), 27.04 (DeSCI), 26.43 (U-net) and **27.84 (BIRNAT)**. Right: Real data from CACTI [23]. The testing time is reported at the bottom row.

for a high-definition video. Two potential problems exist in this conventional sampling plus compression framework: i) the high-dimensional data has to be captured and saved, which requires a significant amount of memory and power; ii) the codec, though efficient, introduces latency for the following transmission. To address the first challenge, one novel idea is to build an *optical encoder*, *i.e.*, compressing the video during capture. Inspired by the compressive sensing (CS) [5, 6], video snapshot compressive imaging (SCI) [13, 23, 36] was proposed aiming to provide a promising solution of this *optical encoder*. The underlying principle is to modulate the video frames with a higher speed than the capture rate of the camera. With knowledge of modulation, high-speed video frames can be reconstructed from each single measurement by using advanced algorithms [53]. It has been shown that 148 frames can be recovered from a single measurement in the coded aperture compressive temporal imaging (CACTI) system [23]. With this optical encoder in hand, another challenge, namely an *efficient decoder* is also required to make the video SCI system being practical. Previous algorithms are usually based on iterative optimization, which needs a long time (even hours [22]) to provide a good result. Inspired by deep learning, there are some research attempting to employ convolutional neural networks (CNNs) to reconstruct the high-speed scene from the SCI measurements [15, 26, 34, 35, 51, 54]. Though the testing speed is promising (tens of milliseconds), none of them can outperform the state-of-the-art optimization algorithm, namely DeSCI [22] on both simulation and real data. Please refer to Fig. 1 for a brief comparison.

Bearing these concerns in mind, in order to achieve high-quality reconstructed videos in a short time, this paper aims to develop an end-to-end deep network to reconstruct high quality images for video SCI, specifically, by investigating the *spatial correlation* via an attention based CNN with Resblock (AttRes-CNN) and *temporal correlation* via a Bidirectional Recurrent Neural Network.

1.1 Video Snapshot Compressive Imaging

As shown in Fig. 2, in video SCI, a dynamic scene, modeled as a time-series of two-dimensional (2D) images, passes through a dynamic aperture which applies

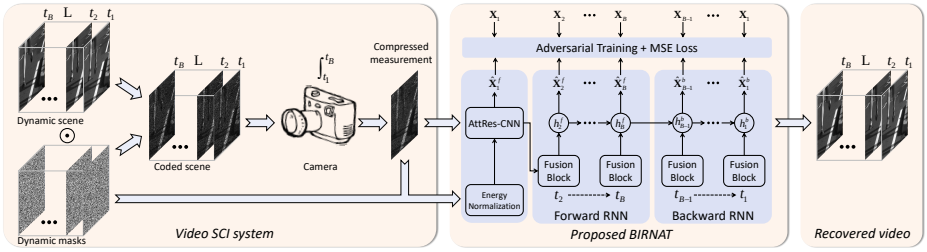

Fig. 2. Principle of video SCI (left) and the proposed BIRNAT for reconstruction (middle). A dynamic scene, shown as a sequence of images at different timestamps ($[t_1$, t_2, ..., $t_B]$, top-left), passes through a dynamic aperture (bottom-left), which imposes individual coding patterns. The coded frames after the aperture are then integrated over time on a camera, forming a single-frame compressed measurement (middle). This measurement along with the dynamic masks are fed into our BIRNAT to reconstruct the time series (right) of the dynamic scene.

timestamp-specified spatial coding. In specific, the value of each timestamp-specified spatial coding is superposed by a random pattern and thus the spatial coding of each two timestamps are different (a shifting binary pattern was used in [23]) from each other. The coded frames after the aperture are then integrated over time on a camera, forming a *compressed coded measurement*. Given the coding pattern for each frame, the time series of the scene can be reconstructed from the compressed measurement through iterative optimization based algorithms, which have been developed extensively before. Based on this idea, different video SCI systems have been built in the literature. The modulation approach can be categorized into spatial light modulator (SLM) (including digital micromirror device (DMD)) [13, 35, 36, 40] and physical mask [23, 55]. However, one common bottleneck to preclude the wide applications of SCI is the *slow reconstruction speed and poor reconstruction quality*. Recently, the DeSCI algorithm, proposed in [22] has led to state-of-the-art results. However, the speed is too slow due to the inherent iterative strategy; it needs about 2 hours to reconstruct eight frames of size 256×256 pixels (Fig. 1 left) from a snapshot measurement, which makes it impractical for real applications.

Motivated by the recent advances of deep learning, one straightforward way is to train an end-to-end network for SCI inversion, with an off-the-shelf structure like U-net [37], which has been used as the backbone of the design for several inverse problems [1, 27, 28, 30, 35]. This was also our first choice but it turned out that a single U-net cannot lead to good results as shown in Fig. 1 since it fails to consider the inherent temporal correlation within in video frames for video SCI. Aiming to fill this research gap, in this paper, we propose a Recurrent Neural Network (RNN) based network dubbed **BI**directional **R**ecurrent **N**eural networks with **A**dversarial **T**raining (BIRNAT) for video SCI reconstruction.

1.2 Related Work

For SCI problems, the well established algorithms include TwIST [2], GAP-TV [52] and GMM [16, 49], where different priors are used. As mentioned before, the DeSCI algorithm [22] has led to state-of-the-art results for video SCI. DeSCI applies the weighted nuclear norm minimization [10] of nonlocal similar patches in the video frames into the alternating direction method of multipliers [3] regime. Inspired by the recent advances of deep learning on image restoration [47, 59], researchers have started using deep learning in computational imaging [15, 21, 26, 30, 35, 48, 57]. Deep fully-connected neural network was used for video C-S in [15] and most recently, a deep tensor ADMM-net was proposed in [26] for video SCI problem. A joint optimization and reconstruction network was trained in [51] for video CS. The coding patterns used in [15] is a repeated pattern of a small block; this is not practical in real imaging systems and only simulation results were shown therein. The real data quality shown in [51] is low. The deep tensor ADMM-net [26] employs deep-unfolding technique [38, 50] and limited results were shown.

To fill the gap of speed and quality for video SCI reconstruction, this paper develop an RNN based network. Intuitively, the desired high-speed video frames are strongly correlated and a network to fully exploit this correlation should improve the reconstructed video quality. RNNs, originally developed to capture temporal correlations for text and speech, e.g., [9, 14], are becoming increasingly popular for video tasks, such as deblurring [32], super-resolution [11, 31] and object segmentation [45]. Although these works achieve high performance in their tasks, how to use RNN to build a unified structure for SCI problems still remains challenging.

1.3 Contributions and Organization of This Paper

In a nutshell, we build a new reconstruction framework (BIRNAT) for video SCI and specific contributions are summarized as follows:

1) We build an end-to-end deep learning based reconstruction regime for video SCI reconstruction and use *RNN* to exploit the temporal correlation.
2) A CNN with Resblock [12] is proposed to reconstruct the first frame as a reference for the reconstruction of following frames by RNN. Considering the limitation of convolution in CNN only extracting the local dependencies, we equip it with a *self-attention* module to capture the global (non-local) spatial dependencies, resulting in AttRes-CNN.
3) Given the reconstruction of the first frame, a *Bidirectional RNN* is developed to sequentially infer the following frames, where the *backward RNN* refines the results of the *forward RNN* to improve the reconstructed video further.
4) This *dual-stage* framework is *jointly trained* via combining mean square error (MSE) loss and *adversarial training* [7] to achieve good results.
5) We apply our model on the six benchmark simulation datasets and it produces 0.59dB higher PSNR than DeSCI on average. We further verify our

BIRNAT on real datasets captured by the CACTI camera and another camera [35]. It shows competitive, sometimes higher performance than DeSCI but with $> 30,000$ times shorter inference time.

The rest of this paper is organized as follows. Sec. 2 present the mathematical model of video SCI. The proposed BIRNAT is developed in Sec. 3. Simulation and real data results are reported in Sec. 4 and Sec. 5 concludes the entire paper.

2 Mathematical Model of Video SCI

Recalling Fig. 2, we assume that B high-speed frames $\{\mathbf{X}_k\}_{k=1}^B \in \mathbb{R}^{n_x \times n_y}$ are modulated by the coding patterns $\{\mathbf{C}_k\}_{k=1}^B \in \mathbb{R}^{n_x \times n_y}$, correspondingly. The measurement $\mathbf{Y} \in \mathbb{R}^{n_x \times n_y}$ is given by

$$\mathbf{Y} = \textstyle\sum_{k=1}^B \mathbf{X}_k \odot \mathbf{C}_k + \mathbf{G}\,, \tag{1}$$

where $\odot$ denotes the Hadamard (element-wise) product and $\mathbf{G}$ represents the noise. For all B pixels (in the B frames) at position (i, j), $i = 1, \ldots, n_x$; $j = 1, \ldots, n_y$, they are collapsed to form one pixel in the snapshot measurement as

$$y_{i,j} = \textstyle\sum_{k=1}^B c_{i,j,k} x_{i,j,k} + g_{i,j}\,. \tag{2}$$

Define $\boldsymbol{x} = \left[\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_B^\top\right]$, where $\boldsymbol{x}_k = \mathrm{vec}(\mathbf{X}_k)$, and let $\mathbf{D}_k = \mathrm{diag}(\mathrm{vec}(\mathbf{C}_k))$, for $k = 1, \ldots, B$, where vec( ) vectorizes the matrix inside ( ) by stacking the columns and diag( ) places the ensured vector into the diagonal of a diagonal matrix. We thus have the vector formulation of the sensing process of video SCI:

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{g}\,, \tag{3}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{n \times nB}$ is the sensing matrix with $n = n_x n_y$, $\boldsymbol{x} \in \mathbb{R}^{nB}$ is the desired signal, and $\boldsymbol{g} \in \mathbb{R}^n$ again denotes the vectorized noise. Unlike traditional CS [5], the sensing matrix considered here is not a dense matrix. In SCI, the matrix $\boldsymbol{\Phi}$ in (3) has a very special structure and can be written as

$$\boldsymbol{\Phi} = [\mathbf{D}_1, \ldots, \mathbf{D}_B]\,, \tag{4}$$

where $\{\mathbf{D}_k\}_{k=1}^B$ are diagonal matrices. Therefore, the compressive sampling rate in SCI is equal to $1/B$. It has recently been proved that high quality reconstruction is achievable when $B > 1$ [18, 19].

3 Proposed Network for Reconstruction

Having obtained the measurement $\mathbf{Y}$ and coding patterns $\{\mathbf{C}_k\}_{k=1}^B$, BIRNAT is developed to *predict* the high-speed frames $\{\widehat{\mathbf{X}}_k\}_{k=1}^B$, which are also regarded as the *reconstructions of real high-speed frames* $\{\mathbf{X}_k\}_{k=1}^B$. In this section, we will introduce each module of the proposed BIRNAT, including a novel *measurement preprocessing* method in Sec. 3.1, an attentional resblock based CNN to reconstruct the first (reference) frame in Sec. 3.2, and a bidirectional RNN to sequentially reconstruct the following frames in Sec. 3.3. Combining adversarial training and MSE loss, BIRNAT is trained end-to-end as described in Sec. 3.4.

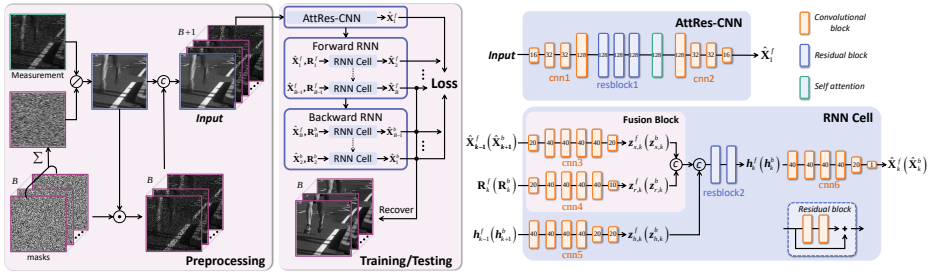

Fig. 3. Left: the proposed preprocessing approach to normalize the measurement. We fed the concatenation of normalization measurement $\overline{\mathbf{Y}}$ and $\{\overline{\mathbf{Y}} \odot \mathbf{C}_k\}_{k=1}^{B}$ into the proposed BIRNAT. Middle: the specific structure of BIRNAT including i) the attention based CNN (AttRes-CNN) to reconstruct the first frame $\widehat{\mathbf{X}}_1^f$; ii) forward RNN to recurrently reconstruct the following frames $\{\widehat{\mathbf{X}}_k^f\}_{k=2}^{B}$; iii) backward RNN to perform reverse-order reconstruction $\{\widehat{\mathbf{X}}_k^b\}_{k=B-1}^{1}$. Right: details of AttRes-CNN and RNN cell. C denotes concatenation along the channel dimension. The numbers in the AttRes-CNN and RNN cell denote the numbers of channels in each feature map.

3.1 Measurement Energy Normalization

Recapping the definition of measurement $\mathbf{Y}$ in (1), it is a weighted ($\{\mathbf{C}_k\}_{k=1}^{B}$) summation of the high-speed frames $\{\mathbf{X}_k\}_{k=1}^{B}$. As a result, $\mathbf{Y}$ is usually a *non-energy-normalized* image. For example, some pixels in $\mathbf{Y}$ may gather only one- or two-pixel energy from $\{\mathbf{X}_k\}_{k=1}^{B}$, while some ones may gather $B-1$ or B. Thus, it is not suitable to directly feed $\mathbf{Y}$ into a network, which motivates us to develop a measurement energy normalization method depicted in Fig. 3 (left).

To be concrete, we first sum all coding patterns $\{\mathbf{C}_k\}_{k=1}^{B}$ to achieve the energy normalization matrix $\mathbf{C}'$ as

$$\mathbf{C}' = \textstyle\sum_{k=1}^{B} \mathbf{C}_k \,, \tag{5}$$

where each element in $\mathbf{C}'$ describes how many corresponding pixels of $\{\mathbf{X}_k\}_{k=1}^{B}$ are integrated into the measurement $\mathbf{Y}$. Then we normalize the measurement $\mathbf{Y}$ by $\mathbf{C}'$ to obtain the energy-normalization measurement $\overline{\mathbf{Y}}$ as

$$\overline{\mathbf{Y}} = \mathbf{Y} \oslash \mathbf{C}', \tag{6}$$

where $\oslash$ denotes the matrix dot (element-wise) division. From Fig. 3 and the definition of $\overline{\mathbf{Y}}$, it can be observed obviously that $\overline{\mathbf{Y}}$ owns more visual information than $\mathbf{Y}$. Meanwhile, $\overline{\mathbf{Y}}$ can be regarded as an approximate average of the high-speed frames $\{\mathbf{X}_k\}_{k=1}^{B}$, preserving the motionless information such as background and motion trail information.

3.2 AttRes-CNN

In order to initiate RNN, a reference frame is required. Towards this end, we propose a ResBlock [12] based deep CNN for the first frame ($\widehat{\mathbf{X}}_1$) reconstruction. Aiming to fuse all the visual information in hand including our proposed

normalization measurement $\overline{\mathbf{Y}}$ and the coding patterns $\{\mathbf{C}_k\}_{k=1}^{B}$, we take the concatenation as:

$$\mathbf{E} = [\overline{\mathbf{Y}}, \overline{\mathbf{Y}} \odot \mathbf{C}_1, \overline{\mathbf{Y}} \odot \mathbf{C}_2, ... \overline{\mathbf{Y}} \odot \mathbf{C}_B]_3 \in \mathbb{R}^{n_x \times n_y \times (B+1)}, \tag{7}$$

where $[\,]_3$ denotes the concatenation along the 3^{rd} dimension. Note that $\{\overline{\mathbf{Y}} \odot \mathbf{C}_k\}_{k=1}^{B}$ are used here to approximate the real mask-modulated frames $\{\mathbf{X}_k \odot \mathbf{C}_k\}_{k=1}^{B}$. After this, $\mathbf{E}$ is fed into a deep CNN (Fig. 3 top-right) consisting two four-layer sub-CNNs ($\mathcal{F}_{cnn1}$ and $\mathcal{F}_{cnn2}$), one three-layer ResBlock ($\mathcal{F}_{resblock1}$), and one self-attention module [43] ($\mathcal{F}_{atten}$) as

$$\widehat{\mathbf{X}}_1 = \mathcal{F}_{cnn2}(\mathbf{L}_3), \;\; \mathbf{L}_3 = \mathcal{F}_{atten}(\mathbf{L}_2), \;\; \mathbf{L}_2 = \mathcal{F}_{resblock1}(\mathbf{L}_1), \;\; \mathbf{L}_1 = \mathcal{F}_{cnn1}(\mathbf{E}), \tag{8}$$

where, $\mathcal{F}_{cnn1}$ is used to fuse different visual information in $\mathbf{E}$ to achieve feature $\mathbf{L}_1$; $\mathcal{F}_{resblock1}$ is employed to further capture the spatial correlation when going deeper, and also to alleviate the gradient vanishing; $\mathcal{F}_{cnn2}$, whose structure is mirror symmetry with $\mathcal{F}_{cnn1}$, is used to reconstruct the first frame $\widehat{\mathbf{X}}_1$ of the desired video, and $\mathcal{F}_{atten}$ is developed to capture long-range dependencies (*e.g.*, non-local similarity), discussed as follows.

Self-attention module. Note that the traditional CNN is only able to capture local dependencies since the convolution operator in CNN has a local receptive field, while in images/videos, non-local similarity [4] is generally used to improve the restoration performance. To explore the non-local information in networks, we employ a self-attention module [44] to capture the long range dependencies [30] among regions to assist our first frame reconstruction.

We perform the self-attention over the pixels of feature map output from $\mathcal{F}_{resblock1}$, denoted by $\mathbf{L}_2 \in \mathbb{R}^{h_x \times h_y \times b}$, where h_x, h_y and b represents the length, width and number of channel in the feature map $\mathbf{L}_2$, respectively. By imposing 1×1 convolution on $\mathbf{L}_2$, we obtain the query $\mathbf{Q}$, key $\mathbf{K}$ and value $\mathbf{V}$ matrix as

$$\mathbf{Q} = \boldsymbol{w}_1 * \mathbf{L}_2, \quad \mathbf{K} = \boldsymbol{w}_2 * \mathbf{L}_2, \quad \mathbf{V} = \boldsymbol{w}_3 * \mathbf{L}_2, \tag{9}$$

where $\{\boldsymbol{w}_1, \boldsymbol{w}_2\} \in \mathbb{R}^{1 \times 1 \times b \times b'}$ and $\boldsymbol{w}_3 \in \mathbb{R}^{1 \times 1 \times b \times b}$ with the fourth dimension representing the number of filters (b' for $\{\boldsymbol{w}_1, \boldsymbol{w}_2\}$ and b for $\boldsymbol{w}_3$), $\{\mathbf{Q}, \mathbf{K}\} \in \mathbb{R}^{h_x \times h_y \times b'}$, $\mathbf{V} \in \mathbb{R}^{h_x \times h_y \times b}$, $*$ represents convolutional operator. $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are then reshaped to $\mathbf{Q}' \in \mathbb{R}^{h_{xy} \times b'}$, $\mathbf{K}' \in \mathbb{R}^{h_{xy} \times b'}$ and $\mathbf{V}' \in \mathbb{R}^{h_{xy} \times b}$, where $h_{xy} = h_x \times h_y$, which means we treat each pixel in the feature map $\mathbf{L}_2$ as a "token", whose feature is $1 \times b'$. After that we construct the attention map $\mathbf{A} \in \mathbb{R}^{h_{xy} \times h_{xy}}$ with element $a_{j,i}$ defined by $a_{j,i} = \frac{exp(s_{i,j})}{\sum_{j=1}^{h_{xy}} exp(s_{i,j})}$, where $s_{i,j}$ is the element in the matrix $\mathbf{S} = \mathbf{Q}'\mathbf{K}'^T \in \mathbb{R}^{h_{xy} \times h_{xy}}$. Here $a_{j,i}$ represents that the extent of the model depends on the i^{th} location when generating the j^{th} region. Having obtained the attention map $\mathbf{A}$, we can impose it on the value matrix $\mathbf{V}'$ to achieve the self-attention feature map $\mathbf{L}_3'$ as

$$\mathbf{L}_3' = reshape(\mathbf{A}\mathbf{V}') \in \mathbb{R}^{h_x \times h_y \times b}, \tag{10}$$

where $reshape()$ reshapes the 2D matrix $\mathbf{AV}' \in \mathbb{R}^{h_{xy} \times b'}$ to the 3D matrix. Lastly, we multiply the self-attention feature map $\mathbf{L}_3'$ by a *scale learnable* parameter λ and add it back to the input feature map $\mathbf{L}_2$ [30], leading to the final result

$$\mathbf{L}_3 = \mathbf{L}_2 + \lambda \mathbf{L}_3'. \tag{11}$$

Recapping the reconstruction process of the first frame in (8), it can be regarded as a nonlinear combination of $\overline{\mathbf{Y}}$ and $\{\overline{\mathbf{Y}} \odot \mathbf{C}_k\}_{k=1}^{B}$. After obtaining the first frame $\widehat{\mathbf{X}}_1$, we use it as a base to reconstruct the following frames by our next proposed sequential model. Therefore, it is important to build the ResBlocks based deep CNN to obtain a good reference frame.

3.3 Bidirectional Recurrent Reconstruction Network

After getting the first frame $\widehat{\mathbf{X}}_1$ via the AttRes-CNN, we now propose a *bidirectional RNN* to perform the reconstruction of the following frames $\{\widehat{\mathbf{X}}_k\}_{k=2}^{B}$ in a sequel manner. The overall structure of BIRNAT is described in Fig. 3, and we give detailed discussion below.

The Forward RNN: The forward RNN takes $\widehat{\mathbf{X}}_1$ as the initial input, fusing different visual information at corresponding frames to sequentially output the forward reconstruction of other frames $\{\widehat{\mathbf{X}}_k^f\}_{k=2}^{B}$ (the superscript f denotes 'forward'). For simplicity, in the following description, we take the frame k as an example to describe the RNN cell, which is naturally extended to each frame.

Specifically, at frame k where $k = 2, \cdots, B$, a fusion block, including two parallel six-layer CNNs $\mathcal{F}_{cnn3}$ and $\mathcal{F}_{cnn4}$, is used to fuse the visual information of the reconstruction at the $(k-1)^{th}$ frame $\widehat{\mathbf{X}}_{k-1}^f$, and a reference image at the k^{th} frame $\mathbf{R}_k$ as

$$\boldsymbol{z}_{i,k}^f = \left[\boldsymbol{z}_{x,k}^f, \boldsymbol{z}_{r,k}^f\right]_3, \quad \boldsymbol{z}_{x,k}^f = \mathcal{F}_{cnn3}(\widehat{\mathbf{X}}_{k-1}^f), \quad \boldsymbol{z}_{r,k}^f = \mathcal{F}_{cnn4}(\mathbf{R}_k^f), \tag{12}$$

where $\widehat{\mathbf{X}}_{k-1}^f$ and $\mathbf{R}_k^f$ are fed into each CNN-based feature extractor to achieve $\boldsymbol{z}_{x,k}^f$ and $\boldsymbol{z}_{r,k}^f$ respectively, which are then concatenated as the fused image feature $\boldsymbol{z}_{i,k}^f$. The reference image at the k^{th} frame, $\mathbf{R}_k^f$, is acquired by

$$\mathbf{R}_k^f = \left[\overline{\mathbf{Y}}, \mathbf{Y} - \textstyle\sum_{t=1}^{k-1} \mathbf{C}_t \odot \widehat{\mathbf{X}}_t^f - \sum_{t=k+1}^{B} \mathbf{C}_t \odot \overline{\mathbf{Y}}\right]_3. \tag{13}$$

Recalling the definition of measurement $\mathbf{Y}$ in (1), the second item in (13) can be seen as an approximation of $\mathbf{C}_k \odot \mathbf{X}_k$. This is due to the reason that the predicted frames $\widehat{\mathbf{X}}_t^f$ before k and our proposed normalization measurement $\overline{\mathbf{Y}}$ after k are used to approximate the corresponding real frames $\mathbf{X}_k$. Basically, considering the approximation of $\widehat{\mathbf{X}}_k^f$ should be more accurate than $\overline{\mathbf{Y}}$, the second item in (13) is going closer to the real $\mathbf{C}_k \odot \mathbf{X}_k$. This is one of the motivation that we build the backward RNN in the following subsection. Furthermore, comparing

the second item on the current frame, the first item $\overline{\mathbf{Y}}$ in (13) contains more consistent visual information over the consecutive frames, such as background. Thus, we put $\overline{\mathbf{Y}}$ in the $\mathbf{R}_k^f$ to help the model reconstruct smoother video frames.

Having obtained $\boldsymbol{z}_{i,k}^f$, it is concatenated with the features $\boldsymbol{z}_{h,k}^f$ extracted from the hidden units $\boldsymbol{h}_{k-1}^f$ (we initialize $\boldsymbol{h}_1$ with zero), to get the fused features $\boldsymbol{g}_k^f$

$$\boldsymbol{g}_k^f = [\boldsymbol{z}_{i,k}^f, \boldsymbol{z}_{h,k}^f]_3, \quad \boldsymbol{z}_{h,k}^f = \mathcal{F}_{cnn5}(\boldsymbol{h}_{k-1}^f), \tag{14}$$

where $\mathcal{F}_{cnn5}$ is another cnn-based feature extractor. After that, $\boldsymbol{g}_k^f$ is fed into a two-layer ResBlock to achieve the hidden units $\boldsymbol{h}_k^f$ at frame k as

$$\boldsymbol{h}_k^f = \mathcal{F}_{resblock2}(\boldsymbol{g}_k^f), \tag{15}$$

which is then used to generate the forward reconstruction $\widehat{\mathbf{X}}_k^f$ by a CNN as

$$\widehat{\mathbf{X}}_k^f = \mathcal{F}_{cnn6}(\boldsymbol{h}_k^f), \tag{16}$$

where $\mathcal{F}_{cnn6}$ is a six-layer CNN. As a result, the current reconstructed frame $\widehat{\mathbf{X}}_k^f$ and hidden units $\boldsymbol{h}_k^f$ are transported to the same cell to sequentially generate the next frame, until we get the last reconstructed frame $\widehat{\mathbf{X}}_B^f$. Finally, we can get the reconstruction of forward RNN $\{\widehat{\mathbf{X}}_k^f\}_{k=1}^B$ (we regard the construction of first frame $\widehat{\mathbf{X}}_1$ from CNN in (8) as $\widehat{\mathbf{X}}_1^f$).

Although the forward RNN is able to achieve appealing results (refer to Table 1), it ignores the sequential information in a reverse order, which has been widely used in natural language processing [17]. Besides, we observe that the performance of forward RNN improves as k goes from 1 to B. We attribute it to the following two reasons: i) the latter frame uses more information from reconstructed frames; ii) the approximation of the second item in (13) are more accurate. Based on these observations, we add the backward RNN to improve the performance of reconstruction further, especially for the front frames.

The Backward RNN: The backward RNN takes $\widehat{\mathbf{X}}_B^f$ and $\boldsymbol{h}_B^f$ as input to sequentially output the backward reconstruction of each frame $\{\widehat{\mathbf{X}}_k^b\}_{k=B-1}^1$ (the superscript b denotes the backward). At frame k, the structure of backward RNN cell is similar to the forward one, with a little difference on the inputs of each cell. Referring to Fig. 3 and the description of the forward RNN above, in the following, we only discuss the difference between backward and forward RNN.

The first difference is the second item in (12). Due to the opposite order to the forward RNN, at frame k, the backward RNN will use the reconstruction of frame $k+1$. The corresponding networks of (12) for backward RNN are thus changed to

$$\boldsymbol{z}_{i,k}^b = [\boldsymbol{z}_{x,k}^b, \boldsymbol{z}_{r,k}^b]_3, \quad \boldsymbol{z}_{x,k}^b = \mathcal{F}_{cnn3}(\widehat{\mathbf{X}}_{k+1}^b), \quad \boldsymbol{z}_{r,k}^b = \mathcal{F}_{cnn4}(\mathbf{R}_k^b). \tag{17}$$

The second difference is the definition of backward reference image $\mathbf{R}_k^b$. According to the definition of $\mathbf{R}_k^f$ in (13) at frame k, since the reconstruction of

frames after k are not obtained, we have to use the normalization measurement $\overline{\mathbf{Y}}$ to approximate them. In backward RNN, it is natural to use each reconstruction from forward RNN $\{\widehat{\mathbf{X}}_k^f\}_{k=1}^B$ directly as

$$\mathbf{R}_k^b = \left[\overline{\mathbf{Y}}, \mathbf{Y} - \sum_{t=B,t\neq k}^{1} \mathbf{C}_t \odot \widehat{\mathbf{X}}_t^f\right]_3, \tag{18}$$

where the first item $\overline{\mathbf{Y}}$ is retained to memory its visual information and help the backward RNN to improve the performance.

The networks used in forward and backward RNN do not share the parameters but have the same structure. Another important difference is that the hidden units $\boldsymbol{h}_1^f$ are set to zeros in the forward RNN, while the hidden units $\boldsymbol{h}_B^b$ are set to $\boldsymbol{h}_B^f$ in the backward RNN. This change builds a closer connection between forward and backward RNN and provides more information for backward RNN.

3.4 Optimization

BIRNAT contains four modules: i) the measurement energy normalization, ii) AttRes-CNN, iii) the forward RNN and iv) the backward RNN. Except for i), other modules have their corresponding parameters. Specifically, all learnable parameters in BIRNAT are denoted by $\boldsymbol{\Theta} = \{\mathbf{W}^c, \mathbf{W}^f, \mathbf{W}^b\}$, where $\mathbf{W}^c = \{\mathbf{W}_{cnn1}^c, \mathbf{W}_{cnn2}^c, \mathbf{W}_{resblock1}^c, \mathbf{W}_{attn}^c\}$ are the parameters of the AttRes-CNN; $\mathbf{W}^f = \{\mathbf{W}_{cnn3}^f, \mathbf{W}_{cnn4}^f, \mathbf{W}_{cnn5}^f, \mathbf{W}_{cnn6}^f, \mathbf{W}_{resblock2}^f\}$ are the parameters of forward RNN; $\mathbf{W}^b = \{\mathbf{W}_{cnn3}^b, \mathbf{W}_{cnn4}^b, \mathbf{W}_{cnn5}^b, \mathbf{W}_{cnn6}^b, \mathbf{W}_{resblock2}^b\}$ are the parameters of backward RNN. In the following, we will introduce how to jointly learn them at the training stage and use the well-learned parameters at the testing stage.

Learning Parameters at the Training Stage. At the training stage, besides measurement and the coding patterns $\{\mathbf{Y}_n, \{\mathbf{C}_{n,k}\}_{k=1}^B\}_{n=1}^N$ for N training videos, the real frames $\{\{\mathbf{X}_{n,k}\}_{k=1}^B\}_{n=1}^N$ are also provided as the supervised signal. In order to minimize the reconstruction error of all the frames, the mean square error is used as the loss function

$$\mathcal{L} = \sum_{n=1}^{N} \alpha \mathcal{L}_n^f + \mathcal{L}_n^b, \tag{19}$$

$$\mathcal{L}_n^f = \sum_{k=1}^{B} ||\widehat{\mathbf{X}}_{n,k}^f - \mathbf{X}_{n,k}||_2^2, \qquad \mathcal{L}_n^b = \sum_{k=B-1}^{1} ||\widehat{\mathbf{X}}_{n,k}^b - \mathbf{X}_{n,k}||_2^2, \tag{20}$$

where $\mathcal{L}_n^f$ and $\mathcal{L}_n^b$ represent the MSE loss of forward and backward RNN, respectively, and α is a trade-off parameter, which is set to 1 in our experiments.

To further improve the quality of each reconstructed frames and make the generated video smoother, we introduce the adversarial training [8] in addition to the MSE loss in (19). To be more specific, the input video frames $\{\mathbf{X}_{n,k}\}_{n=1,k=1}^{N,B}$ are treated as "real" samples, while the reconstructed frames $[\{\widehat{\mathbf{X}}_{n,k}^b\}_{n=1,k=1}^{N,B-1}, \{\widehat{\mathbf{X}}_{n,B}^f\}_{n=1}^N]_3$, generated from previous networks, are assumed as the "fake" samples. The adversarial training loss can be formulated as

$$\mathcal{L}_g = \mathbb{E}_{\mathbf{X}}[\log D(\mathbf{X})] + \mathbb{E}_{\mathbf{Y}}[\log(1 - D(G(\mathbf{Y}, \{\mathbf{C}_k\}_{k=1}^B)))], \tag{21}$$

Table 1. The average results of PSNR in dB (left entry) and SSIM (right entry) and running time per measurement/shot in seconds by different algorithms on 6 datasets. Best results are in **red and bold**, second best results are blue underlined.

Dataset	Kobe		Traffic		Runner		Drop		Aerial		Vehicle		Average		Running time
GAP-TV [52]	26.45	0.8448	20.89	0.7148	28.81	0.9092	34.74	0.9704	25.05	0.8281	24.82	0.8383	26.79	0.8576	4.2
DeSCI [22]	**33.25**	**0.9518**	28.72	0.9250	**38.76**	0.9693	**43.22**	**0.9925**	25.33	0.8603	27.04	0.9094	32.72	0.9347	6180
U-net [35]	27.79	0.8071	24.62	0.8403	34.12	0.9471	36.56	0.9494	27.18	0.8690	26.43	0.8817	29.45	0.8824	0.0312
PnP-FFDNet [54]	30.50	0.9256	24.18	0.8279	32.15	0.9332	40.70	0.9892	25.27	0.8291	25.42	0.8493	29.70	0.8924	3.0
BIRNAT w/o SA&AT&BR	31.06	0.9158	27.17	0.9198	36.62	0.9674	40.67	0.9802	28.40	0.9103	27.24	0.9125	31.86	0.9343	0.0856
BIRNAT w/o SA&AT	32.18	0.9168	28.93	0.9298	38.06	0.9716	42.10	0.9889	28.95	0.9092	27.68	0.9173	32.98	0.9389	0.1489
BIRNAT w/o AT	32.66	0.9490	29.30	0.9418	38.25	0.9748	42.08	0.9914	28.98	0.9163	27.79	0.9234	33.18	0.9494	0.1512
BIRNAT w/o SA	32.27	0.9341	28.99	0.9391	38.44	0.9753	42.22	0.9916	**29.00**	**0.9170**	27.74	0.9233	33.11	0.9467	0.1489
BIRNAT	32.71	0.9504	**29.33**	**0.9422**	38.70	**0.9760**	42.28	0.9918	28.99	0.9166	**27.84**	**0.9274**	**33.31**	**0.9507**	0.1647

where G is the generator which outputs reconstructed video frames, and D is the discriminator that has same structure with [29]. As a result, the final loss function of our model is

$$\mathcal{L} = \sum_{n=1}^{N}(\alpha\mathcal{L}_n^f + \mathcal{L}_n^b) + \beta\mathcal{L}_g, \tag{22}$$

where β is a trade-off parameter. In the experiments, β is set to 0.001.

Performing SCI Reconstruction at the Testing Stage. During testing, with the well-learned network parameters $\boldsymbol{\Theta}$, we can achieve the frames $\{\widehat{\mathbf{X}}_k^f\}_{k=1}^B$ and $\{\widehat{\mathbf{X}}_k^b\}_{k=B-1}^1$. Considering the advantages of backward RNN that uses a good visual features generated by the forward RNN, we use the reconstructed frame 1 to $B-1$ from backward RNN, and frame B from forward RNN to construct the final reconstruction of our system, that is $[\{\widehat{\mathbf{X}}_k^b\}_{k=1}^{B-1}, \widehat{\mathbf{X}}_B^f]_3$. Note that our proposed BIRNAT can also be used in other SCI systems [24, 39, 41, 42, 56, 58].

4 Experiments

In this section, we compare BIRNAT with several state-of-the-art methods on both simulation and real datasets.

4.1 Training, Testing Datasets and Experimental Settings

Datasets: Considering the following two reasons: i) the video SCI reconstruction task does not have a specific training set; ii) the SCI imaging technology is suitable for any scene, we choose the dataset `DAVIS2017` [33], originally used in video object segmentation task as the training set. We first evaluate BIRNAT on six simulation datasets including `Kobe, Runner, Drop, Traffic` [22], `Aerial` and `Vehicle` [26]. After that, we also evaluate BIRNAT on several real datasets captured by real video SCI cameras [23, 35].

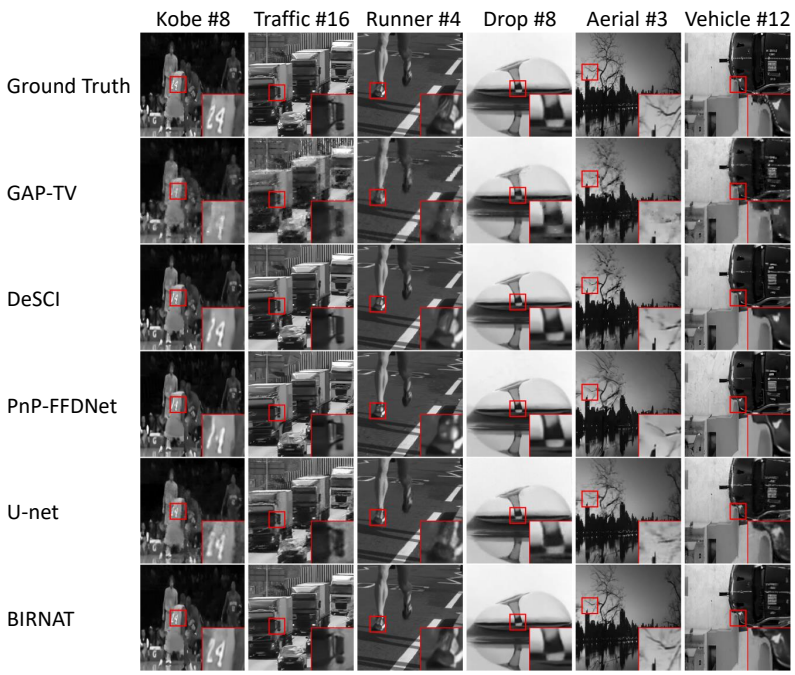

Fig. 4. Reconstructed frames of GAP-TV, DeSCI, U-net and BIRNAT on six simulated video SCI datasets. Please watch the full video in the SM for details.

Implementation Details of BIRNAT: Following the setting in [22], eight ($B = 8$) sequential frames are modulated by the shifting binary masks $\{\mathbf{C}_k\}_{k=1}^{B}$ and then collapsed into a single measurement $\mathbf{Y}$. We randomly crop patch cubes $256 \times 256 \times 8$ from original scenes in `DAVIS2017` and obtain 26,000 training data pairs with data augmentation. Our model is trained for 100 epochs in total. Starting with the initial learning rate of 3×10^{-4}, we reduce the learning rate by 10% every 10 epochs, and it costs about 3 days for training the entire network. The Adam optimizer [20] is employed for the optimization. All experiments are run on the NVIDIA RTX 8000 GPU based on PyTorch. The detailed architecture for BIRNAT is given in the supplementary material (SM).

Counterparts and Performance Metrics: As introduced above, various methods have been proposed for SCI reconstruction. Hereby we compare our model with three competitive counterparts. The first one GAP-TV [52] is a widely used efficient baseline with decent performance. The second one DeSCI [22] currently produces state-of-the-art results. For the results of other algorithms, please refer to [22]. In order to compare with the deep learning based methods, we repurposed U-net to SCI tasks as in [35], where the CNN is employed to capture local correlations in an end-to-end manner. We further compare with the most recent plug-and-play (PnP) algorithm proposed in [54].

For the simulation datasets, both peak-signal-to-noise ratio (PSNR) and structural similarity (SSIM) [46] are used as metrics to evaluate the performance. Besides, to see whether they can be applied to a real-time system, we give the running time of reconstructing the video at the testing stage.

4.2 Results on Simulation Datasets

The performance comparisons on the six benchmark datasets are given in Table 1, using different algorithms, *i.e.* GAP-TV, DeSCI, U-net and various versions of BIRNAT without self-attention (denoted as 'w/o SA') or adversarial training ('w/o AT') or backward RNN ('w/o BR'). It can be observed that: i) BIRNAT outperforms DeSCI on the `Traffic`(0.61dB), `Aerial`(3.66dB) and `Vehicle`(0.80dB) by the metric PSNR. Obviously, BIRNAT can provide superior performance on the datasets with complex background, owning to the non-local features obtained with self-attention and the sequential dependencies constructed by RNN; ii) DeSCI only improved a little bit over BIRNAT on the Kobe(0.54dB), Runner(0.06dB) and Drop(0.94dB), since there are high-speed motions of specfic objects in those three datasets, which are rarely found in the training data. The ADMM-net [26] used different training sets for different testing sets and it only shows the results on `Kobe`(30.15dB), `Aerial`(26.85dB) and `Vehicle`(23.62dB), which are inferior to those of BIRNAT. BIRNAT gets leading average performance on these six datasets both for PSNR and SSIM; iii) BIRNAT achieves 30000 times speedups over DeSCI at the testing stage; iv) the attention mechanism and adversarial training are beneficial to performance.

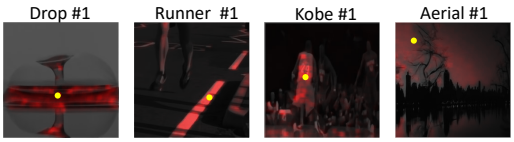

Fig. 5. Selected attention maps of the first frame. Yellow points denote the pixels randomly selected from each image, and red areas denote the active places.

Fig. 4 shows selected reconstructed frames of BIRNAT on these six datasets compared with GAP-TV, DeSCI and our repurposed U-net. We can observe that while DeSCI smooths out the details in the reconstructed video, BIRNAT provides sharper borders and finer details, owning to the better interpolation with both spatial and temporal information extracted by CNN and Bidirectional RNN. To further explore the influence of attention mechanism, we illustrate the attention map in Fig. 5, where we plot the attended active areas (highlighted red color) of a randomly selected pixel. It can be seen that those non-local regions in red color are corresponding to the highly semantically related areas. These attention-aware features can provide long range spatial dependencies among pixels, which is helpful for the first frame reconstruction and gives a better basement for generating the following frames.

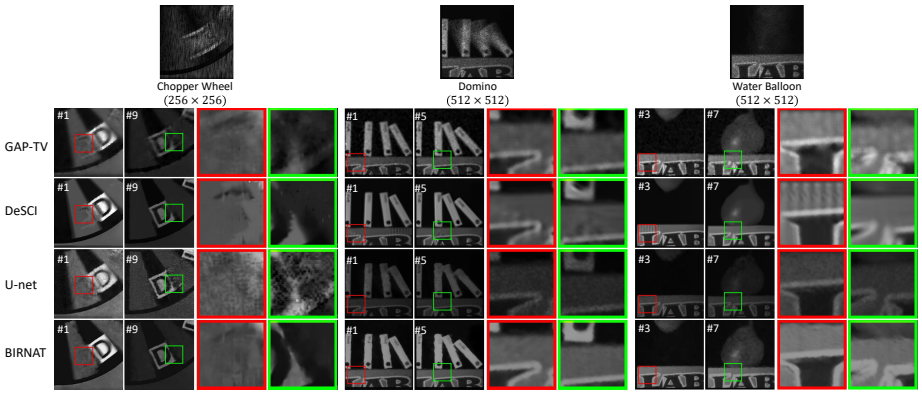

Fig. 6. Real data `Wheel`(left), `Domino` (middle) and `Water Balloon` (right): results of GAP-TV, DeSCI, U-net and BIRNAT.Please refer to more real data results in the SM.

4.3 Results on Real SCI Data

We now apply BIRNAT to real data captured by the SCI cameras [23, 35] to verify the its robustness. The `Wheel` snapshot measurement of size 256×256 pixels encodes 14 ($B = 14$) high-speed videos. The mask is the shifting random mask with the pixel shifts determined by the pre-set translation of the printed film. The `Domino` and `Water Balloon` snapshot measurement of size 512×512 pixels encodes 10 frames, in which the mask is controlled by a DMD [35]. The real captured data have noise inside and thus the SCI for real data is more challenging. As shown in Fig. 6, the reconstructed video by BIRNAT shows finer and complete details compared with other methods, with a significant saving on the reconstruction time during testing compared to DeSCI. This indicates the applicability and efficiency of our algorithm in real applications.

5 Conclusions

In this paper, we have proposed a bidirectional RNN with adversarial training for snapshot compressive imaging system, called BIRNAT. We employ a dual-stage framework, where the first frame is reconstructed through an attention ResBlock based deep CNN, and then the following frames are sequentially inferred by RNN. The experimental results on both simulation and real-world SCI camera data have demonstrated that the proposed method achieves superior performance and outperforms current state-of-the-art algorithms.

Acknowledgement

B. Chen acknowledges the support of the Program for Oversea Talent by Chinese Central Government, the 111 Project (No. B18039), and NSFC (61771361) and Shaanxi Innovation Team Project.

References

1. Barbastathis, G., Ozcan, A., Situ, G.: On the use of deep learning for computational imaging. Optica **6**(8), 921–943 (2019)
2. Bioucas-Dias, J., Figueiredo, M.: A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration. IEEE Transactions on Image Processing **16**(12), 2992–3004 (December 2007)
3. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning **3**(1), 1–122 (January 2011)
4. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. vol. 2, pp. 60–65. IEEE (2005)
5. Donoho, D.L.: Compressed sensing. IEEE Transactions on Information Theory **52**(4), 1289–1306 (April 2006)
6. Emmanuel, C., Romberg, J., Tao, T.: Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. IEEE Transactions on Information Theory **52**(2), 489–509 (February 2006)
7. Goodfellow, I.: Nips 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160 (2016)
8. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. pp. 2672–2680. NIPS'14 (2014)
9. Graves, A., Mohamed, A., Hinton, G.: Speech recognition with deep recurrent neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 6645–6649 (May 2013). https://doi.org/10.1109/ICASSP.2013.6638947
10. Gu, S., Zhang, L., Zuo, W., Feng, X.: Weighted nuclear norm minimization with application to image denoising. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2862–2869 (2014)
11. Haris, M., Shakhnarovich, G., Ukita, N.: Recurrent back-projection network for video super-resolution. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
12. He, K., Zhang, X., Ren, S., J, S.: Deep residual learning for image recognition. In: CVPR (2016)
13. Hitomi, Y., Gu, J., Gupta, M., Mitsunaga, T., Nayar, S.K.: Video from a single coded exposure photograph using a learned over-complete dictionary. In: 2011 International Conference on Computer Vision. pp. 287–294. IEEE (2011)
14. Huang, Y., Wang, W., Wang, L.: Video super-resolution via bidirectional recurrent convolutional networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(4), 1015–1028 (April 2018). https://doi.org/10.1109/TPAMI.2017.2701380
15. Iliadis, M., Spinoulas, L., Katsaggelos, A.K.: Deep fully-connected networks for video compressive sensing. Digital Signal Processing **72**, 9–18 (2018). https://doi.org/10.1016/j.dsp.2017.09.010
16. J. Yang, ., X. Yuan, ., X. Liao, ., P. Llull, ., G. Sapiro, ., Brady, D.J., Carin, L.: Video compressive sensing using Gaussian mixture models. IEEE Transaction on Image Processing **23**(11), 4863–4878 (November 2014)

17. Jaeger, H.: A tutorial on training recurrent neural networks, covering bppt, rtrl, ekf and the "echo state network" approach (2005)
18. Jalali, S., Yuan, X.: Snapshot compressed sensing: Performance bounds and algorithms. IEEE Transactions on Information Theory **65**(12), 8005–8024 (Dec 2019). https://doi.org/10.1109/TIT.2019.2940666
19. Jalali, S., Yuan, X.: Compressive imaging via one-shot measurements. In: IEEE International Symposium on Information Theory (ISIT) (2018)
20. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
21. Kulkarni, K., Lohit, S., Turaga, P., Kerviche, R., Ashok, A.: Reconnet: Non-iterative reconstruction of images from compressively sensed random measurements. In: CVPR (2016)
22. Liu, Y., Yuan, X., Suo, J., Brady, D., Dai, Q.: Rank minimization for snapshot compressive imaging. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(12), 2990–3006 (Dec 2019)
23. Llull, P., Liao, X., Yuan, X., Yang, J., Kittle, D., Carin, L., Sapiro, G., Brady, D.J.: Coded aperture compressive temporal imaging. Optics Express **21**(9), 10526–10545 (2013). https://doi.org/10.1364/OE.21.010526
24. Llull, P., Yuan, X., Carin, L., Brady, D.J.: Image translation for single-shot focal tomography. Optica **2**(9), 822–825 (2015)
25. Lu, G., Ouyang, W., Xu, D., Zhang, X., Cai, C., Gao, Z.: Dvc: An end-to-end deep video compression framework. CVPR (2019)
26. Ma, J., Liu, X., Shou, Z., Yuan, X.: Deep tensor admm-net for snapshot compressive imaging. In: IEEE/CVF Conference on Computer Vision (ICCV) (2019)
27. Meng, Z., Ma, J., Yuan, X.: End-to-end low cost compressive spectral imaging with spatial-spectral self-attention. In: European Conference on Computer Vision (ECCV) (August 2020)
28. Meng, Z., Qiao, M., Ma, J., Yu, Z., Xu, K., Yuan, X.: Snapshot multispectral endomicroscopy. Opt. Lett. **45**(14), 3897–3900 (Jul 2020)
29. Mescheder, L., Nowozin, S., Geiger, A.: Which training methods for gans do actually converge? In: International Conference on Machine Learning (ICML) (2018)
30. Miao, X., Yuan, X., Pu, Y., Athitsos, V.: λ-net: Reconstruct hyperspectral images from a snapshot measurement. In: IEEE/CVF Conference on Computer Vision (ICCV) (2019)
31. Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., Khudanpur, S.: Recurrent neural network based language model. In: INTERSPEECH. vol. 2, p. 3 (2010)
32. Nah, S., Son, S., Lee, K.M.: Recurrent neural networks with intra-frame iterations for video deblurring. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
33. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbelaez, P., Sorkine-Hornung, A., Gool, L.V.: The 2017 DAVIS challenge on video object segmentation. CoRR **abs/1704.00675** (2017), http://arxiv.org/abs/1704.00675
34. Qiao, M., Liu, X., Yuan, X.: Snapshot spatial–temporal compressive imaging. Opt. Lett. **45**(7), 1659–1662 (Apr 2020)
35. Qiao, M., Meng, Z., Ma, J., Yuan, X.: Deep learning for video compressive sensing. APL Photonics **5**(3), 030801 (2020). https://doi.org/10.1063/1.5140721, https://doi.org/10.1063/1.5140721
36. Reddy, D., Veeraraghavan, A., Chellappa, R.: P2c2: Programmable pixel compressive camera for high speed imaging. In: CVPR 2011. pp. 329–336. IEEE (2011)
37. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI). LNCS, vol. 9351, pp. 234–241. Springer (2015),

http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a, (available on arXiv:1505.04597 [cs.CV])
38. Roux, J.R.L., Weninger, J.: Deep unfolding: Model-based inspiration of novel deep architectures (2014)
39. Sun, Y., Yuan, X., Pang, S.: High-speed compressive range imaging based on active illumination. Optics Express **24**(20), 22836–22846 (Oct 2016)
40. Sun, Y., Yuan, X., Pang, S.: Compressive high-speed stereo imaging. Opt Express **25**(15), 18182–18190 (2017). https://doi.org/10.1364/OE.25.018182
41. Tsai, T.H., Llull, P., Yuan, X., Carin, L., Brady, D.J.: Spectral-temporal compressive imaging. Optics Letters **40**(17), 4054–4057 (Sep 2015)
42. Tsai, T.H., Yuan, X., Brady, D.J.: Spatial light modulator based color polarization imaging. Optics Express **23**(9), 11912–11926 (May 2015)
43. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 5998–6008. Curran Associates, Inc. (2017), http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf
44. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems. pp. 5998–6008 (2017)
45. Ventura, C., Bellver, M., Girbau, A., Salvador, A., Marques, F., Giro-i Nieto, X.: Rvos: End-to-end recurrent network for video object segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
46. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., et al.: Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2004)
47. Xie, J., Xu, L., Chen, E.: Image denoising and inpainting with deep neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 341–349. Curran Associates, Inc. (2012), http://papers.nips.cc/paper/4686-image-denoising-and-inpainting-with-deep-neural-networks.pdf
48. Xu, K., Ren, F.: CSVideoNet: A real-time end-to-end learning framework for high-frame-rate video compressive sensing. arXiv: 1612.05203 (Dec 2016)
49. Yang, J., Liao, X., Yuan, X., Llull, P., Brady, D.J., Sapiro, G., Carin, L.: Compressive sensing by learning a Gaussian mixture model from measurements. IEEE Transaction on Image Processing **24**(1), 106–119 (January 2015)
50. Yang, Y., Sun, J., Li, H., Xu, Z.: Deep admm-net for compressive sensing mri. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29, pp. 10–18. Curran Associates, Inc. (2016)
51. Yoshida, M., Torii, A., Okutomi, M., Endo, K., Sugiyama, Y., Taniguchi, R.i., Nagahara, H.: Joint optimization for compressive video sensing and reconstruction under hardware constraints. In: The European Conference on Computer Vision (ECCV) (September 2018)
52. Yuan, X.: Generalized alternating projection based total variation minimization for compressive sensing. In: 2016 IEEE International Conference on Image Processing (ICIP). pp. 2539–2543 (Sept 2016)
53. Yuan, X., Brady, D., Katsaggelos, A.K.: Snapshot compressive imaging: Theory, algorithms and applications. IEEE Signal Processing Magazine (2020)

54. Yuan, X., Liu, Y., Suo, J., Dai, Q.: Plug-and-play algorithms for large-scale snapshot compressive imaging. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
55. Yuan, X., Llull, P., Liao, X., Yang, J., Brady, D.J., Sapiro, G., Carin, L.: Low-cost compressive sensing for color video and depth. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3318–3325 (2014). https://doi.org/10.1109/CVPR.2014.424
56. Yuan, X., Pang, S.: Structured illumination temporal compressive microscopy. Biomedical Optics Express **7**, 746–758 (2016)
57. Yuan, X., Pu, Y.: Parallel lensless compressive imaging via deep convolutional neural networks. Optics Express **26**(2), 1962–1977 (Jan 2018)
58. Yuan, X., Tsai, T.H., Zhu, R., Llull, P., Brady, D., Carin, L.: Compressive hyperspectral imaging with side information. IEEE Journal of Selected Topics in Signal Processing **9**(6), 964–976 (September 2015)
59. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Transactions on Image Processing **26**(7), 3142–3155 (July 2017). https://doi.org/10.1109/TIP.2017.2662206