Robust Deep Compressive Sensing with Recurrent-Residual Structural Constraints

Jun Niu

Abstract—Existing deep compressive sensing (CS) methods either ignore adaptive online optimization or depend on costly iterative optimizer during reconstruction. This work explores a novel image CS framework with recurrent-residual structural constraint, termed as R²CS-NET. The R²CS-NET first progressively optimizes the acquired samplings through a novel recurrent neural network. The cascaded residual convolutional network then fully reconstructs the image from optimized latent representation. As the first deep CS framework efficiently bridging adaptive online optimization, the R²CS-NET integrates the robustness of online optimization with the efficiency and nonlinear capacity of deep learning methods. Signal correlation has been addressed through the network architecture. The adaptive sensing nature further makes it an ideal candidate for color image CS via leveraging channel correlation. Numerical experiments verify the proposed recurrent latent optimization design not only fulfills the adaptation motivation, but also outperforms classic long short-term memory (LSTM) architecture in the same scenario. The overall framework demonstrates hardware implementation feasibility, with leading robustness and generalization capability among existing deep CS benchmarks.

Index Terms—Compressive sensing, structural constraint, adaptive online optimization, recurrent neural network.

I. INTRODUCTION

Information compression and reconstruction stands at the center of communication and signal processing. Compressive sensing (CS) has been an emerging strategy to recover original signals with considerably fewer measurements at sub-Nyquist sampling rate [1], [2]. Different from auto-encoders, where both encoder and decoder can be heavily engineered, conventional CS seeks to alleviate the sampling complexity while reconstructing signals from low-dimensional measurements via computationally intensive optimization. Its sampling efficiency accommodates limited sensor-side computational resources and transmission bandwidth. Compared with deploying complex encoder at the frontend, CS serves as a more economic and realistic option. The online optimization further offers extra robustness to amend measurement noise during reconstruction. Benefiting from its flexible and efficient architecture, CS has been successfully applied in scenarios where measurements are expensive and noisy. Typical applications include magnetic resonance imaging (MRI) and high resolution imaging systems [3], [4].

Classic CS theory assumes the signal to be sparse or can be expressed as a sparse linear combination of known basis. This strong requirement hinders CS’s practical application as such basis is implicit in general. Recently, Bora et al. introduces deep learning based structural constraint to relax this requirement [5]. A pre-trained deep neural network (NN) is proposed as the structural constraint in the place of sparsity, which suffices a generalized set-restricted eigenvalue condition. With deep structural constraint and successful training, low reconstruction error for dense signals can still be achieved with high probability. The reconstruction, however, heavily relies on classic iterative optimization. Its slow convergence remains a critical challenge in practice. In addition, the sampling relies on random measurement matrices, which is sub-optimal for highly structured signals such as natural images.

Inspired by deep NN’s powerful ability in signal representation, several deep CS methods are proposed to enhance the performance of both sampling and reconstruction. For example, Wu S. et al. [6] proposed an unrolled feed-forward NN for learning the measurement matrix, while depending on classic iterative optimization for reconstruction. Kulkarni et al. proposed a non-iterative deep reconstruction model, the ReconNet [7]. It achieves high efficiency and robustness with convolutional neural network (CNN), though the architecture does not emphasize domain insights. The DR²-NET [8] applies residual learning for reconstruction enhancement. The MS-DCI [9] develops an elegant multi-scale sampling framework with cascaded multi-stage reconstruction. The multi-level sampling decomposition, however, potentially increases the sampling hardware complexity or transmission bandwidth. More recent frameworks, such as OPINE-NET [10] and AMP-NET [11], unfold iterative optimization algorithms for model based methods into networks. Though efficient and interpretable, the unrolled networks possess reduced robustness and generalization capacity versus the iterative online optimizers. As an alternative, recently Wu Y. et al. [4] combines explicit online optimization with learnable deep structural constraint and measurement matrix. This framework largely speeds up the convergence of online optimization. However, it still relies on the classic iterative gradient descent optimizer for adaptive optimization, which is computationally costly and slow.

This work proposes a novel framework, termed as R²CS-NET, to effectively bridge deep image CS with adaptive online optimization. In particular, a novel recurrent neural network (RNN) is explored as an alternative for unrolled NN during initial reconstruction. It aims to enhance the structural constraint’s capacity, such that the overall system operates more robustly and better generalizes in online usage, while avoiding expensive operations like iterative gradient calculation. The R²CS-NET is composed of three stages: measurement, recurrent latent optimization, and residual convolutional reconstruction. The measurement stage assembles a learnable, modified structural random matrix [12]. It is the only module deployed.
at the sensor frontend, and exploits switchable pre-coded masks to adapt the R²-CS-NET in different sampling domains without re-train necessity. The recurrent latent optimization attempts to acquire the robustness and generalization capacity of online optimization. This stage is established through a novel recurrent pipeline. It progressively optimizes the latent representation of the acquired samplings to accommodate the downstream reconstruction network. The residual CNN then fully reconstructs the original image in pixel domain. The recurrent latent optimization network and cascaded residual CNN together serve as the structural constraint in the place of sparsity. All components are fully convolutional for concurrency acceleration. The three stages are optimized jointly during training toward optimal reconstruction quality under given sampling rate.

Signal correlations have been addressed through the network architecture. The adaptive sensing nature of R²-CS-NET further makes it an ideal candidate for color image CS. Numerical experiments verify the efficiency and robustness of the proposed design. The R²-CS-NET also demonstrates leading performance among recent deep CS benchmarks.

In summary, this work contributes the following:
- The R²-CS-NET is the first deep CS framework bridges efficient online optimization with unrolled reconstruction network. Expensive operations such as gradient computations are completely avoided during reconstruction.
- A novel recurrent-residual architecture is explored to enhance the generalization capacity and robustness of deep structural constraints.
- The R²-CS-NET intrinsically supports color image CS. The overall system exhibits leading performance among deep CS benchmarks.

II. DEEP COMPRESSIVE SENSING WITH STRUCTURAL CONSTRAINTS AND ONLINE OPTIMIZATION

Compressive sensing aims to efficiently acquire a compressible signal \( x \) and then reconstruct the original signal from samplings \( m \) via finding optimal solutions to the under-determined linear equation system. Exploiting signal’s sparsity or incoherence nature, the original signal can be recovered with far fewer samplings required by the Nyquist–Shannon sampling theorem. Formally,

\[
m = Fx + \eta,
\]

where \( F \) is the measurement matrix and \( \eta \) denotes the noise at measurement time. Depending on the sparsity of original signal \( x \), the dimension of measured signal \( m \) is usually considerably smaller than that of \( x \).

A necessary and sufficient condition for this problem to be well conditioned is the Restricted Isometry Property (RIP). It states, for a small constant \( \delta \in (0, 1) \), the difference between two signals \( x_1 - x_2 \) satisfies

\[
(1 - \delta)||x_1 - x_2||_2^2 \leq ||F(x_1 - x_2)||_2^2 \leq (1 + \delta)||x_1 - x_2||_2^2.
\]

In other words, the measurement matrix \( F \) preserves the distance between original sparse signals with small perturbation [4]. The original signal thus can be recovered by solving the optimization problem

\[
\hat{x} = \arg\min_x \|m - Fx\|_2^2.
\]

Classic CS theory requires \( x \) to be a sparse signal or a dense signal that can be expressed as a sparse linear combination of known basis vectors. This enforces a strong restriction. In most practical problems the sparse basis vectors are not explicit to discover, sometimes extremely untrivial to obtain.

As an alternative, Bora et al. [5] propose using a pre-trained deep NN as the structural constraint in the place of sparsity

\[
x = G(z).
\]

The network \( G(\cdot) \) implicitly constrains the compressibility through its architecture, where the latent space tensor \( z \)'s dimension is usually considerably smaller than that of \( x \). This condition is sufficient to provide a generalized Set-Restricted Eigenvalue Condition (S-REC). It guarantees that low reconstruction error can be achieved with high probability.

Instead of directly solving Eq. [3], the reconstruction now first estimates the latent representation \( z \) by solving the online optimization problem

\[
\hat{z} = \arg\min_z ||m - FG(z)||_2^2,
\]

then reconstructs the original signal through \( \hat{x} = G(\hat{z}) \).

The structural constrained CS framework connects deep learning with classic CS theory. It provides a theoretical foundation to tackle arbitrary dense, compressible signals in data-driven manner. A few deep CS methods have been proposed thereafter. For example, the reconstruction can be accomplished through either classic optimizers or unrolled NN designs; the structural constraint \( G \) can be learnt together with the measurement matrix to achieve better reconstruction quality [4], [10], [13]-[15]. However, whenever adaptive online optimization is leveraged during the reconstruction, the gradient-based optimization of Eq. [5] has to be addressed in every iteration. As the structural constraint typically consists of heavy-weight NN, repeated evaluation of \( \frac{\partial G(z)}{\partial z} \) along the iterative optimization turns out to be extremely expensive. The computational cost further increases with more complex architecture designs. The unrolled NN reconstruction sidesteps the cost concerns. However, ignoring the adaptive online optimization potentially leads to sub-optimal solution with non-perfect \( G(\cdot) \), impairing the system’s robustness under the inevitable measurement noise in practice. An efficient, robust deep CS solution remains at the center of the problem.

III. R²-CS-NET: A NOVEL IMAGE COMPRESSIVE SENSING FRAMEWORK

This work introduces a novel framework to efficiently bridge deep CS with adaptive online optimization. The proposed R²-CS-NET consists of three main stages: measurement, recurrent latent optimization, and residual convolutional reconstruction. The measurement stage is engineered with simplicity and hardware implementation feasibility. With acquired samplings, an RNN first progressively adapts the samplings’ latent
representation to the downstream reconstruction network. The residual CNN then fully reconstructs the original image. The recurrent latent optimization and the residual CNN together serve as the recurrent-residual structural constraint. It delivers higher generalization capacity and robustness than unrolled networks at moderate latency.

Each stage has its specific motivation. However, the three objectives are not mutually independent. All stages can be cascaded into an end-to-end system during training. A system sketch is illustrated in Fig. 1. The network architecture exploits signal correlation for optimal sampling efficiency.

IV. MEASUREMENT

The measurement stage is the only module deployed with sensors. One primary difference between CS and auto-encoders is the emphasis on efficiency and simplicity of data acquisition. The measurement stage is therefore engineered with hardware compatibility in mind. It consists of two fully convolutional layers, one with switchable pre-coded kernels adapting the R2CS-NET from various data acquisition spaces to the preferred sampling space; the other operating as the standard measurement matrix. The measurement matrix is the only component to be learnt from data.

The first layer’s operation can be written as

$$\mathbf{Y}^{[1]} = \mathbf{W}^{[1],\text{mask}}_M \ast \mathbf{X},$$

where $\ast$ denotes the 2D convolution operator; $\mathbf{X}$ is the image in acquisition domain; $\mathbf{W}^{[1],\text{mask}}_M$ is the adaptive coded mask. Convolution stride is chosen as $[N_1, N_2]$ along width and height dimension, respectively; $N_1$ and $N_2$ denotes the width and height of each sampling block. This layer has no trainable parameters. The parallelism nature of convolution operations further guarantees concurrency efficiency. Figure 2 depicts the mapping mechanism.

Without loss of generality, we consider pixel domain sampling as the default setting. The front layer maps pixel value into k-space to exploit natural images’ spectral domain sparsity. Inspired by this operation’s discrete cosine transform (DCT) nature, the following set of coded masks are plugged in place of convolutional kernels.

$$\mathbf{W}^{\text{DCT}}_{i,j,k+k_1,N_{ch},+k_2,N_{ch}} = \frac{2}{N_1} \frac{2}{N_2} \alpha(k_1)\alpha(k_2) \cos \left( \frac{2n_1 + 1}{N_1} \pi k_1 \right) \cos \left( \frac{2n_2 + 1}{N_2} \pi k_2 \right),$$

$$i = 0, 1, ..., N_1 - 1; j = 0, 1, ..., N_2 - 1;$$

$$k = 0, ..., N_{ch} - 1,$$

where $N_{ch}$ denotes the colorspace channel. The measurement then multi-scales on frequency bands of input images. This step largely relieves the complexity expectation of the followed operation, which now can solely focus on linear measurement without further domain transformation. When directly acquiring data in k-space, one can simply replace the above with a set of binary coded masks [16]. The entire R2CS-NET thereafter operates in the new sampling space without re-train necessity. Other acquisition spaces can be adapted accordingly.

The second convolutional layer samples the original image block by block, across image’s channels

$$\mathbf{M} = \mathbf{W}^{[2]}_M \ast \mathbf{Y}^{[1]},$$

where $\mathbf{W}^{[2]}_M$ is the learnable CS measurement matrix; $\mathbf{M}$ is the tensor of measured samplings. The convolution kernel is set to be of size $[1, 1]$ with stride 1 for block-wise measurement. Each tensor slice in $\mathbf{M}$ with unit width and unit height thus represents the measurements in one sampling block.

One observation is the above two layers assemble a modified, learnable structural random matrix (SRM) [12]. The SRM is defined as a product of matrices $\Phi = \mathbf{DTR}$, where $\mathbf{D}$ is the subsampling matrix; $\mathbf{T}$ is the transform kernel; $\mathbf{R}$ is a randomizer. $\mathbf{W}^{[1]}_M$ and $\mathbf{W}^{[2],\text{mask}}_M$ perform in the place of $\mathbf{D}$ and $\mathbf{T}$, respectively.

The measurement stage can be feasibly integrated on hardware with coded aperture cameras [17] or CS-CMOS cameras [18]. For example, the switchable coded mask $\mathbf{W}^{[1],\text{mask}}_M$ and the learnt measurement matrix $\mathbf{W}^{[2]}_M$ can be implemented as high resolution coded apertures modulating the optical field from the scene, with a low resolution focal plane aperture instated for sampling data acquisition. Only the coded aperture needs to be switched when operating pre-trained R2CS-NET in new sampling space. The computationally inexpensive coded mask and measurement matrix can also be realized in digital domain as on CS-CMOS cameras, whose on-chip processor is strictly restricted by limited power and area on sensors.

V. RECURRENT-RESIDUAL STRUCTURAL CONSTRAINT

The recurrent-residual structural constraint aims to bridge the advantages of adaptive online optimization with deep CS, while avoiding the intensive costs of classic iterative optimizers. A two-phase layout is proposed to this end. The recurrent latent optimization stage resembles the gradient descent optimization through a novel RNN architecture. It preliminarily updates the samplings in latent space. The cascaded residual CNN fully reconstructs the image thereafter. Compared with the unrolled NN methods, the recurrent-residual structural constraint better generalizes on input patterns. Though not citing particular noise prior, it expands the robustness under measurement noise through architecture design with successful data-driven training.

A. Recurrent Latent Representation Optimization

The latent representation can be generally of arbitrary shape. Aiming at preliminarily reconstructing toward the original image, here $z$ is designed with the same dimension as $\mathbf{Y}^{[1]}$. The gradient descent latent optimization is thus formulated as

$$\hat{\mathbf{Z}}_{i+1} = \hat{\mathbf{Z}}_i - \alpha \frac{\partial}{\partial \hat{\mathbf{Z}}_i} \| F(\hat{\mathbf{Z}}_i) - \mathbf{M} \|^2_2,$$

where the subscript $i$ denotes the iteration step; $\hat{\mathbf{Z}}_{i+1}$ is progressively updated through the incremental adjustment to $\hat{\mathbf{Z}}_i$. 
Fig. 1: Systematic sketch of the R$^2$CS-NET. The system consists of three stages: measurement, adaptive recurrent optimization, and residual convolutional reconstruction. Only the measurement stage needs to be deployed with frontend sensor hardware.

Fig. 2: Mapping input images from acquisition domain to k-space. The \( W_M^{[1,\text{mask}]} \) kernel is a set of 64 x \( n_{ch} \) coded masks.

Our key observation is the stochastic gradient descent update of Eq. (9) resembles the cell state update in classic RNNs [19].

\[
C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \quad (10)
\]

where the subscript \( t \) denotes the recurrent step; \( -i_t \) operates as the learning rate and the new candidate state \( \tilde{C}_t \) corresponds to the incremental adjustment; \( f_t \) operates as the reset coefficient for the previous state; \( \odot \) denotes the Hadamard product.

This encourages us to design a recurrent unit assembling similar online latent optimization. Each iteration in Eq. (9) explicitly depends on the previous output \( Z_i \) and the original measurements \( \mathcal{M} \). This is robust in clean sampling environment. However, with the presence of measurement noise, the optimization deviates from the reconstruction goal due to the noisy \( \mathcal{M} \). We consider a similar RNN. At each step, the recurrent unit consumes \( \tilde{Z}_i \) and \( F(\tilde{Z}_i) - \mathcal{M} \) as inputs. In noisy sampling environment both \( F(\tilde{Z}_i) \) and \( \mathcal{M} \) are perturbed. Their absolute difference along the iterations converges towards a non-zero random variable. This residual potentially indicates environmental attributes such as noise variance, and assists the RNN amending the latent representation.

The second intuition is the wavelet decomposition nature of measurement residuals. The linear measurement operation maintains the residuals in the same space along the optimization steps. This is comparable to a wavelet sequence. One transform that successfully updates one packet is highly likely to successfully update the next packet in the same space.

A novel RNN is proposed based on the above analysis to progressively optimize the samplings’ latent representation through measurement residuals. At each step, the recurrent unit operates upon the residual between the original samplings and measurements of previous step's output. The residual eventually converges to zero in noise-free environment. In noisy measurement environment, it highlights the noise attributes for further denoising amendment. This architecture together with successful data-driven training serves as the prior that enhances the system’s noise robustness. One practical concern, however, is the measurement residuals vary in magnitudes over the recurrence. The recurrent unit is expected to possess consistent performance over wide bandwidth. To tackle this challenge, a scaling gate network renders the residual input with appropriate gain before generating the new candidate. \( i_t \) and \( f_t \) are both designed as CNNs to further enhance the recurrent unit’s capacity.

Summarily, at the initial step, the input \( I_0 \) is initialized with the measurement sampling \( \mathcal{M} \). At each following step \( i \), the recurrent unit updates the latent representation \( Z_i \) as

\[
I_i = S(F(Z_{i-1}) - I_0),
\]

\[
H_i = C(Z_i),
\]

\[
Z_i = R(Z_{i-1}, H_i) \odot Z_{i-1} + U(Z_{i-1}, H_i) \odot H_i,
\]

where \( S(\cdot), C(\cdot), R(\cdot), U(\cdot) \) are all CNNs instead of coefficients; \( F(\cdot) \) is the measurement operation following Eq. (8). The scaling gate \( S(\cdot) \) adjusts the residual input with appropriate gain at each step. The candidate gate \( C(\cdot) \) generates a new candidate from scaled input. The latent representation is then updated as a trade-off between previous result and new candidate, where the reset gate \( R(\cdot) \) and the update gate \( U(\cdot) \) calculate the trade-off coefficients, respectively.

Conventional LSTM and GRU cells use dense connection architecture. This contradicts with the uncorrelated nature of samplings measured from different image blocks, simultaneously resulting in numerous unnecessary parameters. As an alternative, \( S(\cdot), C(\cdot), R(\cdot), U(\cdot) \) consist of 2D convolution layers with kernel size [1,1] and stride 1. Operating on the latent representation sampling block by sampling block, it successfully avoids block-wise coupling. A batch normalization layer is further added at the recurrent input and output to enhance nonlinear processing capability.

Figure 3-4 depicts the proposed recurrent unit architecture and computation pipeline, respectively. At the initial step the candidate gate directly operates on the measured samplings \( \mathcal{M} \). Latent representation then receives progressive updates from previous step’s output and measurement residual. All network parameters are shared across the recurrent steps.
addresses the inter-channel correlation while maintaining the

\[ \mathcal{X} = G^{[2]}(\mathcal{Z}) + \mathcal{X}_1 \]  

where \( \mathcal{X} \) is as close to the original image \( \mathcal{X} \) as possible. Sharpness enhancement and artifacts suppression need to be accomplished toward this goal.

Successful reconstruction relies on comprehending the intrinsic correlation of \( \mathcal{Z} \)'s components. Here we first enrich the latent representation through cascaded convolutions

\[ G^{[l]}(\mathcal{Z}) = BN(\mathbf{W}_G^{[l]} \ast G^{[l-1]}(\mathcal{Z}) + \mathbf{B}_G^{[l]}), l = 1, 2 \]  

where \( BN(\cdot) \) denotes a batch normalization layer; superscript \([l]\) denotes the layer index. \( \mathbf{W}_G^{[1]} \) and \( \mathbf{W}_G^{[2]} \) contain \( N_1 \times N_2 \times N_{ch} \) channels. The block-wise latent optimization addresses the inter-channel correlation while maintaining the relative independency across adjacent sampling blocks. As a result, both kernels are correspondingly designed as size \([1, 1]\) with stride 1.

A transposed convolution layer then maps \( G^{[3]}(\mathcal{Z}) \) to the original image’s shape

\[ \hat{G}^{[3]}(\mathcal{H}) = \mathbf{C}_G^{[3]T} \cdot \hat{G}^{[2]}(\mathcal{Z}) + \mathbf{B}_G^{[3]}, \]  

where \( \cdot \) denotes the matrix multiplication; \( \mathbf{C}_G^{[3]} \) is the matrix operator of the transposed convolution; \( \hat{G}^{[2]}(\cdot) \) and \( \hat{G}^{[3]}(\cdot) \) denote the row-major order vector representation of the tensor outputs of \( G^{[2]}(\cdot) \) and \( G^{[3]}(\cdot) \), respectively. After this operation, the latent representation is transformed to be in same shape as \( \mathcal{X} \), while maintaining the same number of entries.

The initial macroblocking artifacts suppression is achieved with three cascading convolution layers with the same operations described by Eq. (12). The transposed convolution maps entries along tensor channels to width and height dimensions. As a result, the convolution kernels of shape \([11, 11], [7, 7]\) and \([1, 1]\) are used to exploit the intra-channel spatial correlation of \( G^{[3]}(\mathcal{Z}) \). To guarantee the initial decoding results falling into the non-negative pixel value range, a rectifier linear unit (ReLU) function activates the output of \( G^{[6]} \).

With macroblocking artifacts mitigated, detail enhancement remains the next objective towards high quality reconstruction. Here we apply cascaded residual CNNs for image sharpening. Operations in each residual block can be formulated as

\[ C^{[l]}(\mathcal{H}) = \begin{cases} \mathbf{W}_G^{[l]} \ast G^{[l-1]}(\mathcal{H}) + \mathbf{B}_G^{[l]} + G^{[l-3]}(\mathcal{H}), & l = 9, 12, 15, 18, 21 \\ ReLU(BN(\mathbf{W}_G^{[l]} \ast G^{[l-1]}(\mathcal{H}) + \mathbf{B}_G^{[l]})), & l = 7, 8, \ldots, 20 \end{cases} \]  

This architecture conceptually decomposes the image into wavelet series. The convolution layers inside each residual block focuses on sharpening high frequency details. After merging with the low frequency image snapshots, features of the complete image are delivered to the next cascaded residual block for further enhancement.

Passing through five cascaded residual convolution blocks, the final reconstruction is activated with a ReLU function:

\[ \hat{\mathcal{X}} = ReLU(G^{[21]}(\mathcal{H})). \]  

It enforces pixel values into the non-negative definition range, and enhances the model’s nonlinear capacity in the meantime.

VI. LEARNING MEASUREMENT MATRIX AND STRUCTURAL CONSTRAINT

Successful learning fulfills each of the \( \mathbb{R}^2 \)-CS-NET’s three stages its particular role. An effective measurement stage enforces the RIP condition while collaborating with the reconstruction stages for optimal sampling efficiency. This objective can be described through the measurement loss \([4]\)

\[ L_M = \frac{1}{N} \sum_{i=1}^{N} (||F(\hat{\mathcal{X}}_i - \mathcal{X}_i)||_2^2 - ||\hat{\mathcal{X}}_i - \mathcal{X}_i||_2^2)^2, \]  

where \( N \) is the number of training images.
Meanwhile, the recurrent latent optimization stage adapts the samplings to accommodate the downstream reconstruction. Given a trained downstream network, this objective is outlined by the loss function

\[ L_R = \frac{1}{N} \sum_{i=1}^{N} \| F \cdot G(\hat{Z}_i) - F(X_i) \|_2^2, \]  

(17)

where \( \hat{Z}_i \) denotes the optimized latent representation of samplings; \( G(\hat{Z}_i) \) reconstructs \( X_i \).

The residual convolutional stage minimizes the discrepancy between final reconstructions and the ground truth. One common formulation is the mean squared error (MSE) loss

\[ L_{MSE} = \frac{1}{N} \sum_{i=1}^{N} \| \hat{X}_i - X_i \|_2^2. \]  

(18)

It equivalently optimizes the reconstructed images’ peak signal-to-noise ratio (PSNR) \([16],[20]\). However, according to specific engineering applications, the loss function can be easily revised to address other perceptual quality metrics \([21]\).

Each stage can be trained separately toward its goal or iteratively updated in interaction. However, despite the distinct motivations, the three objectives are not mutually independent. \( L_R \) and \( L_{MSE} \) simultaneously reach their minimum with optimized discrepancy between final reconstruction \( \hat{X} \) and the ground truth \( X \). Cascading as an end-to-end system during training, all stages can be jointly trained under \( L_{MSE} \). This is applied as a preferred training strategy realizing it not only avoids potential slow convergence, but also enforces the optimal sampling - reconstruction of the overall system.

\section{VII. Experiments}

This section provides intensive experiments to validate the efficiency and performance of the proposed \( R^2 \)CS-NET framework. Without loss of generality, we set the recurrent optimization length as \( T = 20 \) and sampling block size as \( 8 \times 8 \times N_{ch} \). These hyper-parameters are empirically selected with accommodating the limited computational resources. Though not optimized through intensive hyper-parameter search, the \( R^2 \)CS-NET shows leading performance among existing deep CS benchmarks. The advantage of the proposed recurrent-residual constraint is demonstrated through its enhancement to reconstruction quality and noise robustness under various measurement noise levels. Comparison with classic online optimization methods demonstrates the \( R^2 \)CS-NET achieves comparable sensing fidelity at considerably higher efficiency.

\subsection{A. Training}

The training and validation datasets are cropped from the DIV2K dataset \([22]\), which consists of 1,000 high quality color images of 2K resolution. Without loss of generality, we set the dimension of training images as \( 128 \times 128 \times 3 \), where the last dimension denotes the images’ RGB channels in pixel domain. The \( R^2 \)CS-NET framework, however, is insensitive to the image size. The entire training dataset contains 1,098,400 images at \( 128 \times 128 \) resolution and the validation dataset has 1000 images at the same resolution. The training, validation and testing datasets throughout the discussion share no overlap.

Assembled as an end-to-end system, all modules are updated iteratively by minimizing Eq. (18) with Adam optimizer. The recurrent-residual structural constraint enhances the generalization capacity in architecture. This potential needs to be further attained through appropriate data-driven training. An environmental noise is thus engineered to instill the robustness under practical application. Throughout the training process, a white Gaussian noise with standard deviation \( \sigma = 0.1 \) is added to all measurement operations during sampling and recurrent latent optimization. This perturbation is empirically chosen to the extent that no noticeable reconstruction deterioration is observed in noise-free environment. Once training completes, the \( R^2 \)CS-NET can reconstruct images at arbitrary resolution and operate under arbitrary noise level.

\subsection{B. Validating the Recurrent Latent Optimization}

We first validate the recurrent latent optimization design with ablation study. Sampling and reconstructing the Lena image in Set14, Fig. 5 illustrates the samplings’ latent representation at various recurrent steps. The latent features are progressively optimized to accommodate the downstream reconstruction. Evolution and convergence can be easily observed along the snapshots. Figure 6 demonstrates the reconstructed image at each optimization step, evaluated under PSNR and structural similarity index measure (SSIM) metrics. The quality evolution aligns with the progressive optimization principle and successfully verifies the architecture design.

![Fig. 5: Optimized latent representation \( Z_i \) at recurrent step \( i = 5, 10, 15, 20 \), illustrating the tensor’s first channel.]

\subsection{C. Sampling Efficiency and Reconstruction Quality}

The efficiency of learnable measurement matrix, the effectiveness of the residual CNN reconstruction, and the contribution of recurrent latent optimization are further validated through lateral comparison with control group counterparts.
| Sampling Rate | Model         | Dataset | LIVE1 | Set5 | Set14 | BSD300 | BSD500 | Time Cost on LIVE1 | Trainable Parameters |
|--------------|--------------|---------|-------|------|-------|--------|--------|--------------------|---------------------|
| 0.02         | Random-RCS   | 24.48   | 0.748 | 23.95 | 0.813 | 22.35  | 0.743  | 23.20              | 0.740               |
|              | RCS          | 24.40   | 0.794 | 26.85 | 0.874 | 24.65  | 0.802  | 25.13              | 0.787               |
|              | RCS-GD-20    | 24.46   | 0.796 | 26.98 | 0.876 | 24.74  | 0.804  | 25.21              | 0.789               |
|              | RCS-GD-100   | 24.52   | 0.798 | 27.10 | 0.879 | 24.82  | 0.806  | 25.27              | 0.792               |
|              | R²CS-LSTM    | 24.43   | 0.794 | 26.82 | 0.873 | 24.63  | 0.801  | 25.14              | 0.786               |
|              | R²CS-NET     | 24.54   | 0.794 | 26.99 | 0.878 | 24.78  | 0.805  | 25.25              | 0.790               |
| 0.05         | Random-RCS   | 27.11   | 0.882 | 29.91 | 0.927 | 27.22  | 0.876  | 27.73              | 0.885               |
|              | RCS          | 27.21   | 0.885 | 30.09 | 0.929 | 27.35  | 0.881  | 27.82              | 0.877               |
|              | RCS-GD-20    | 27.31   | 0.887 | 30.29 | 0.932 | 27.48  | 0.884  | 27.92              | 0.881               |
|              | RCS-GD-100   | 27.31   | 0.887 | 30.29 | 0.932 | 27.48  | 0.884  | 27.92              | 0.881               |
|              | R²CS-LSTM    | 27.25   | 0.885 | 30.05 | 0.929 | 27.39  | 0.882  | 27.75              | 0.876               |
|              | R²CS-NET     | 27.50   | 0.889 | 30.33 | 0.933 | 27.59  | 0.885  | 27.97              | 0.880               |
| 0.10         | Random-RCS   | 27.43   | 0.892 | 29.74 | 0.924 | 27.16  | 0.880  | 27.83              | 0.884               |
|              | RCS          | 29.87   | 0.935 | 32.42 | 0.952 | 29.74  | 0.925  | 30.38              | 0.931               |
|              | RCS-GD-20    | 30.08   | 0.938 | 32.81 | 0.956 | 29.98  | 0.927  | 30.59              | 0.934               |
|              | RCS-GD-100   | 30.09   | 0.938 | 32.81 | 0.956 | 29.98  | 0.927  | 30.59              | 0.934               |
|              | R²CS-LSTM    | 30.31   | 0.940 | 32.79 | 0.957 | 30.09  | 0.929  | 30.72              | 0.935               |
|              | R²CS-NET     | 30.31   | 0.940 | 32.79 | 0.957 | 30.09  | 0.929  | 30.72              | 0.935               |
| 0.20         | Random-RCS   | 30.66   | 0.946 | 32.45 | 0.951 | 29.69  | 0.922  | 31.25              | 0.944               |
|              | RCS          | 33.97   | 0.973 | 35.35 | 0.970 | 32.82  | 0.955  | 35.44              | 0.974               |
|              | RCS-GD-20    | 34.33   | 0.975 | 35.82 | 0.973 | 33.15  | 0.958  | 35.14              | 0.976               |
|              | RCS-GD-100   | 34.75   | 0.978 | 36.43 | 0.976 | 33.54  | 0.960  | 35.62              | 0.979               |
|              | R²CS-LSTM    | 34.67   | 0.978 | 35.90 | 0.974 | 33.47  | 0.961  | 35.34              | 0.978               |
|              | R²CS-NET     | 34.79   | 0.978 | 35.97 | 0.974 | 33.56  | 0.961  | 35.54              | 0.978               |
| 0.30         | Random-RCS   | 37.30   | 0.974 | 39.25 | 0.974 | 38.90  | 0.941  | 39.59              | 0.974               |
|              | RCS          | 37.78   | 0.987 | 38.22 | 0.981 | 35.65  | 0.970  | 38.67              | 0.989               |
|              | RCS-GD-20    | 38.68   | 0.990 | 39.33 | 0.984 | 36.45  | 0.974  | 38.97              | 0.991               |
|              | RCS-GD-100   | 38.78   | 0.989 | 39.39 | 0.988 | 35.97  | 0.973  | 39.10              | 0.990               |
|              | R²CS-LSTM    | 38.90   | 0.991 | 39.07 | 0.984 | 38.90  | 0.991  | 39.57              | 0.992               |
|              | R²CS-NET     | 38.90   | 0.991 | 39.07 | 0.984 | 38.90  | 0.991  | 39.57              | 0.992               |

*Figures 7 and [7] shows the reconstructions of four methods on Barbara in Set14, from both clean and noisy measurement environment. The R²CS-NET achieves comparable fidelity as the RCS-GD-100, prevailing the methods absenting online optimization. The recurrent-residual structural constraint introduces moderate latency in adaptive optimization. However, decoupled from sampling, it has no direct impact on measurement latency or sensor hardware complexity.*

Each of the following architectures is investigated on NVIDIA GeForce RTX 2080 Super GPU with the same runtime setting.

- **Random-RCS:** samples with fixed, classic random measurement matrix; reconstructs via the residual CNN without online optimization.
- **RCS:** samples with the learnable measurement matrix, with remaining parts the same as the Random-RCS.
- **RCS-GD:** employs the gradient descent adaptive online optimization upon the RCS, following Eq. (5). Iteration steps of 20 and 100 are investigated in the experiment, where convergence is empirically reached at 100 steps.
- **R²CS-LSTM:** employs the classic LSTM architecture for recurrent latent optimization upon the RCS. The LSTM unit passes measurement residuals in the same way as the proposed recurrent-residual stage.
- **R²CS-NET:** the proposed framework.

Five groups of model instances are trained at 0.02, 0.05, 0.1, 0.2 and 0.3 sampling rate, respectively. Their statistical performance is evaluated on five widely used benchmark datasets: LIVE1 [23], Set5 [22], Set14 [22], and the test sets of BSD300 [24], BSD500 [25].

Table I summarizes the performance of each experimental group. The Random-RCS serves as the base case. Its samplings are acquired through random measurement matrix with no online optimization. The successful reconstruction, meanwhile, demonstrates the fundamental validity of the residual convolutional reconstruction stage. Comparison between the Random-RCS and the RCS reveals the sampling efficiency delivered by the learnable measurement matrix. The CS methods with online optimization, namely the RCS-GD, R²CS-LSTM, and R²CS-NET, demonstrate better performance in general. The consistently superior reconstruction of the RCS-GD over the RCS indicates the significance of online optimization. Despite the RCS-GD’s leading CS fidelity, its classic gradient descent online optimizer is extremely slow. The R²CS-NET employs the proposed recurrent latent optimization stage to achieve comparable reconstruction quality with considerably higher efficiency. Its time efficiency and slightly better fidelity over the RCS-GD-100 attest the effectiveness. As a control group, the R²CS-LSTM utilizes the classic convolutional LSTM unit for adaptive optimization. The proposed recurrent architecture in the R²CS-NET outperforms the LSTM in the same scenario. Though the R²CS-NET contains more parameters, the two methods operate at comparable speed.
TABLE II: Model Comparison on the Color BSD500 Test Set, Using Averaged PSNR (dB) and SSIM Metrics (1st & 2nd Column under each Sampling Rate).

| Dataset       | Model      | Sampling Rate | 0.1       | 0.25      | 0.3       |
|---------------|------------|---------------|-----------|-----------|-----------|
| BSD500 Test Set | ReconNet 17 | 26.85 0.784 30.51 0.897 31.49 0.917 |
|               | CSNet+ 29  | 29.07 0.820 33.26 0.925 34.52 0.945 |
|               | LDIT 25    | 24.94 0.704 29.10 0.852 30.23 0.880 |
|               | LDAMP 28   | 26.61 0.692 28.35 0.830 29.89 0.872 |
|               | DPDNN 31   | 24.37 0.686 28.87 0.849 29.98 0.876 |
|               | NN 50     | 23.44 0.644 26.42 0.776 27.23 0.804 |
|               | ISTA-NET 92 | 25.46 0.502 29.92 0.738 32.21 0.769 |
|               | AMP-NET 11 | 28.11 0.810 32.55 0.923 33.82 0.941 |
|               | R²CS-NET 17 | 30.89 0.943 37.56 0.987 39.73 0.992 |

E. Comparing with Deep CS Benchmarks

Verified the R²CS-NET’s validity via stage-by-stage lateral comparisons, this section further investigates the overall framework’s performance amidst recent deep CS benchmarks. The R²CS-NET enlists its efficacy through learnable sampling and efficient adaptive reconstruction. The correlations across and within image channels are carefully addressed at both measurement and reconstruction stages. Additionally assisted with recurrent latent optimization, the R²CS-NET’s adaptive sensing nature makes it a natural candidate for color image CS. Table II first compares it with existing benchmarks on BSD500. Noticing most existing CS benchmarks are reported on grayscale images, a set of R²CS-NET instances are further trained with single channel images. Table III inspects the candidates on Set11 [14], containing 11 grayscale images. Among the benchmarks, TVAL3 [26], DAMP [27] are classic model-based CS methods. LDIT [28], LDAMP [28] are based on random measurement, while the remaining methods generally support learning measurement matrix through data. ReconNet [7] and CSNet+ [29] employ classic deep network architecture, while AMP-NET [11], LDIT, LDAMP, NN [30], DPDNN [31], ISTA-NET+ [32] unfold the iterative optimization into feedforward networks. CSNet+, LDIT, LDAMP fully address color images’ channel correlation. R²CS-NET is the first method integrates adaptive online optimization with unrolled network.

The adaptive online optimization as well as the R²CS-NET’s overall architecture successfully handles sampling correlation, contributing toward outstanding performance in color image CS. The margin over benchmark methods expands along with the sampling rate. Leading performance can also be observed on grayscale test. Figure 10 visualizes reconstructions on Parrots image in Set11 at 0.1 sampling rate. All methods sample pixel domain images except the R²CS-NETk. As a validation of the framework’s sampling adaptivity, the R²CS-NETk transplants the R²CS-NET for k-space sampling by merely switching the measurement mask. It achieves the same outcome except for inevitable rounding errors.

VIII. CONCLUSIONS AND FUTURE WORK

This work proposes a novel deep CS framework, termed as R²CS-NET. Inspired by iterative online optimizers, a recurrent-residual structural constraint is explored for image CS. As an alternative for unrolled NN, it possess improved robustness and generalization capacity while maintaining high
Fig. 7: Reconstructions on Barbara image in Set14 at sampling rate of 0.1. Highlighted regions are enlarged in the 2nd row.

Fig. 8: Comparison between R²CS-NET and benchmarks under various measurement noise levels at 0.1 sampling rate.

The R²CS-NET demonstrates leading CS quality and measurement noise robustness among existing benchmarks. Designs inspired by more advanced optimization models can be further investigated in specific application domains.

References

[1] R. G. Baraniuk, “Compressive sensing [lecture notes],” IEEE Signal Processing Magazine, vol. 24, no. 4, pp. 118–121, 2007.

[2] E. J. Candès and M. B. Wakin, “An introduction to compressive sampling,” IEEE Signal Processing Magazine, vol. 25, no. 2, pp. 21–30, 2008.

[3] J. C. Ye, “Compressed sensing mri: a review from signal processing perspective,” BMC Biomedical Engineering, vol. 1, no. 1, p. 8, Mar 2019.

[4] Y. Wu, M. Rosca, and T. Lillicrap, “Deep compressed sensing,” in Proceedings of the 36th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 6850–6860.

[5] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, “Compressed sensing using generative models,” in Proceedings of the 34th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 537–546.

[6] S. Wu, A. Dimakis, S. Sanghavi, F. Yu, D. Holtmann-Rice, D. Storcheus, A. Rostamizadeh, and S. Kumar, “Learning a compressed sensing measurement matrix via gradient unrolling,” in Proceedings of the 36th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 6828–6839.

[7] S. Lohit, K. Kulkarni, R. Kerviche, P. Turaga, and A. Ashok, “Convolutional neural networks for noniterative reconstruction of compressively sensed images,” IEEE Transactions on Computational Imaging, vol. 4, no. 3, pp. 326–340, 2018.

[8] H. Yao, F. Dai, S. Zhang, Y. Zhang, Q. Tian, and C. Xu, “Dr2-net: Deep residual reconstruction network for image compressive sensing,” Neurocomputing, vol. 359, pp. 483–493, 2019.

[9] T. N. Canh and B. Jeon, “Multi-scale deep compressive imaging,” IEEE Transactions on Computational Imaging, vol. 7, pp. 86–97, 2021.

[10] J. Zhang, C. Zhao, and W. Gao, “Optimization-inspired compact deep compressive sensing,” IEEE Journal of Selected Topics in Signal Processing, vol. 14, no. 4, pp. 765–774, 2020.

[11] Z. Zhang, Y. Liu, J. Liu, F. Wen, and C. Zhu, “Amp-net: Denoising-based deep unfolding for compressive image sensing,” IEEE Transactions on Image Processing, vol. 30, pp. 1487–1500, 2021.

[12] T. T. Do, L. Gan, N. H. Nguyen, and T. D. Tran, “Fast and efficient com-
Fig. 9: $R^2$CS-NET and RCS reconstruction on *Monarch* image in Set14 at sampling rate of 0.1, under various measurement noise levels. Highlighted regions are enlarged in the 2nd row.

Fig. 10: Reconstructions on *Parrots* image in Set11 at 0.1 sampling rate. $R^2$CS-NET$_k$ acquires samplings in k-space; all others sample from pixel domain.
[32] J. Zhang and B. Ghanem, “Ista-net: Interpretable optimization-inspired deep network for image compressive sensing,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 1828–1837.