Optimization Inspired Network Design for Image Reconstruction

Jian Zhang

IVUL Meeting

January 26, 2017

Focus

- Y. Yang, J. Sun, H. Li, Z. Xu, *Deep ADMM-Net for Compressive Sensing MRI*, NIPS 2016, Spain, Dec. 5-10.
- From Xi’an Jiaotong University, China

Contents

- Regularization-based Framework for Image Reconstruction
  - Denoising, Inpainting, Deblurring, Compressive Sensing
  - Regularization-based Framework
  - Optimization (Proximal Splitting, Iterated Shrinkage-Thresholding)
- Deep Convolutional Neural Network for Image Reconstruction
  - Super-resolution
  - Image Denoising
- **Deep ADMM-Net for Compressive Sensing MRI**

Image Reconstruction

Process：

Sampling → Compression → Transmission → Storage

Image Reconstruction

- Deblurring

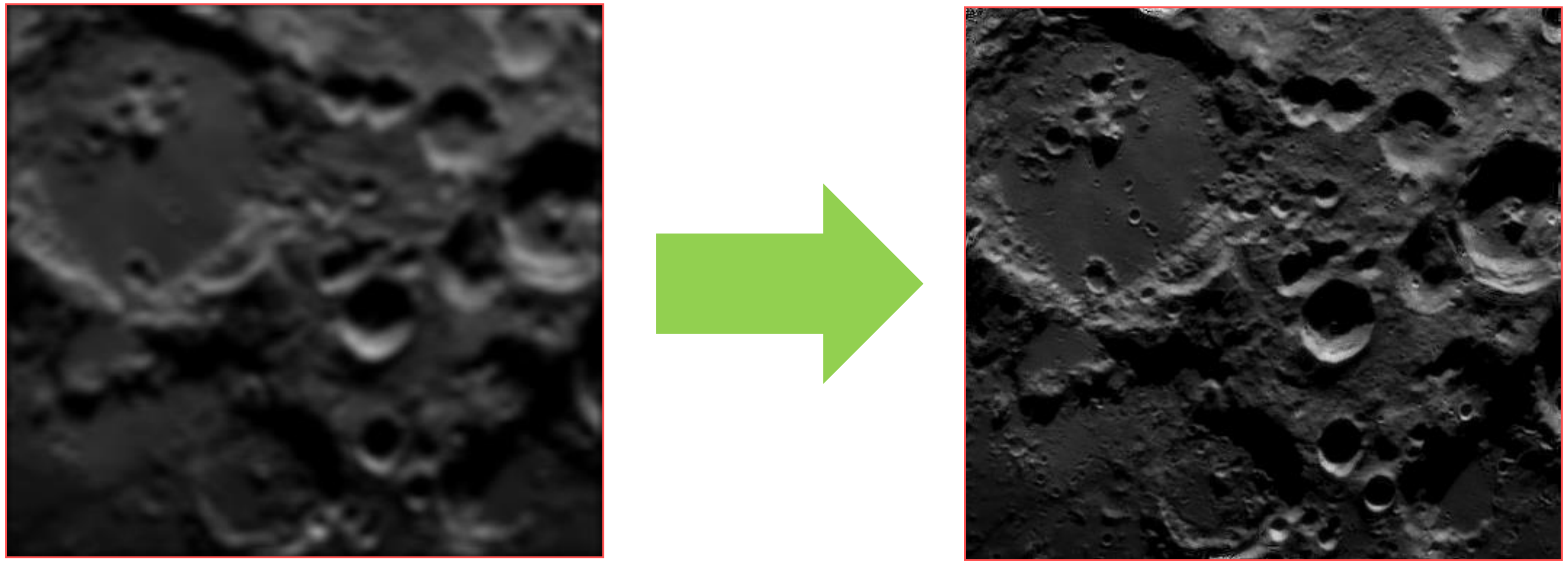


Image Reconstruction

- Inpainting

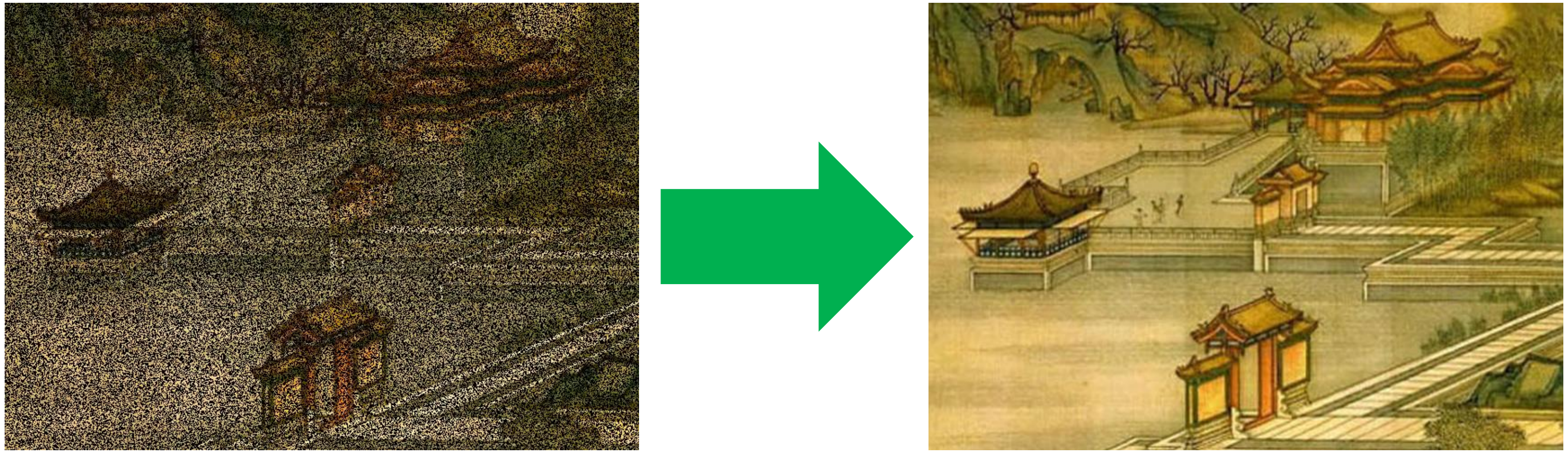


Image Reconstruction

- Magnetic Resonance Imaging Compressive Sensing (MRI-CS)

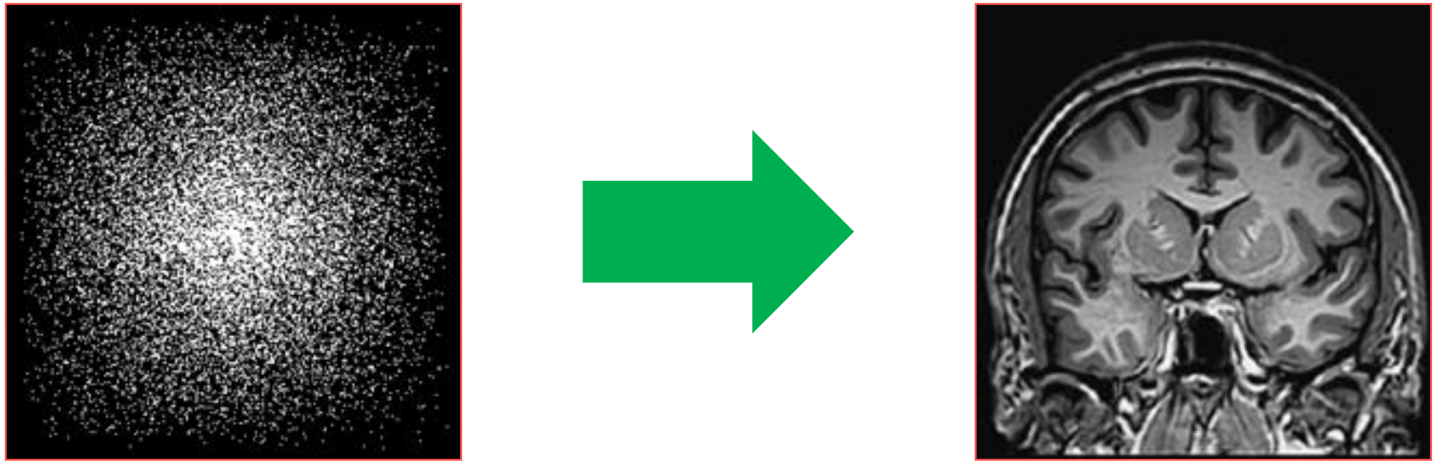


Image Reconstruction

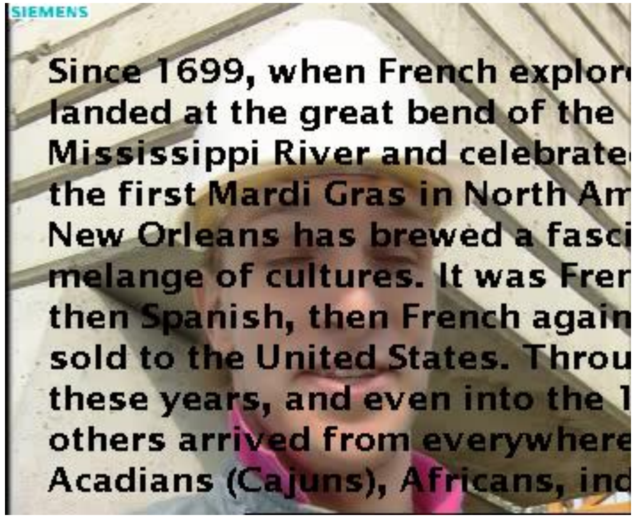
- Super-resolution

Image Reconstruction

- Super-resolution

Image Reconstruction

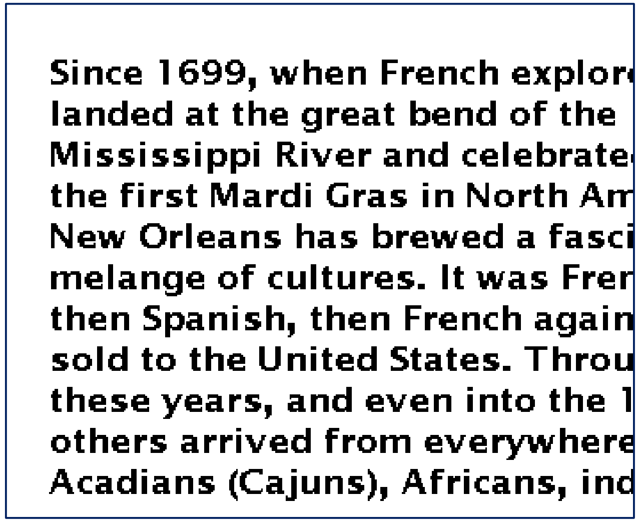
- Inpainting



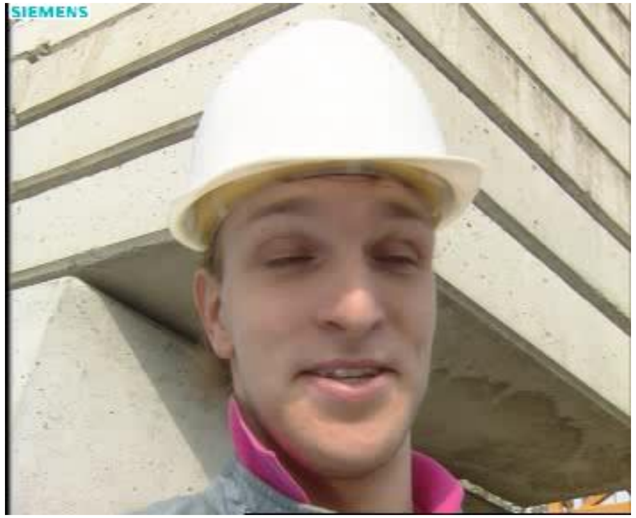
$\boldsymbol{y}$

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n}$$

$\boldsymbol{H}$



$\otimes$



$\boldsymbol{x}$

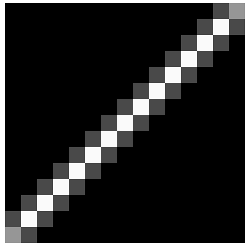
Image Reconstruction

- Deblurring

$\boldsymbol{y}$

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n}$$

$\boldsymbol{H}$

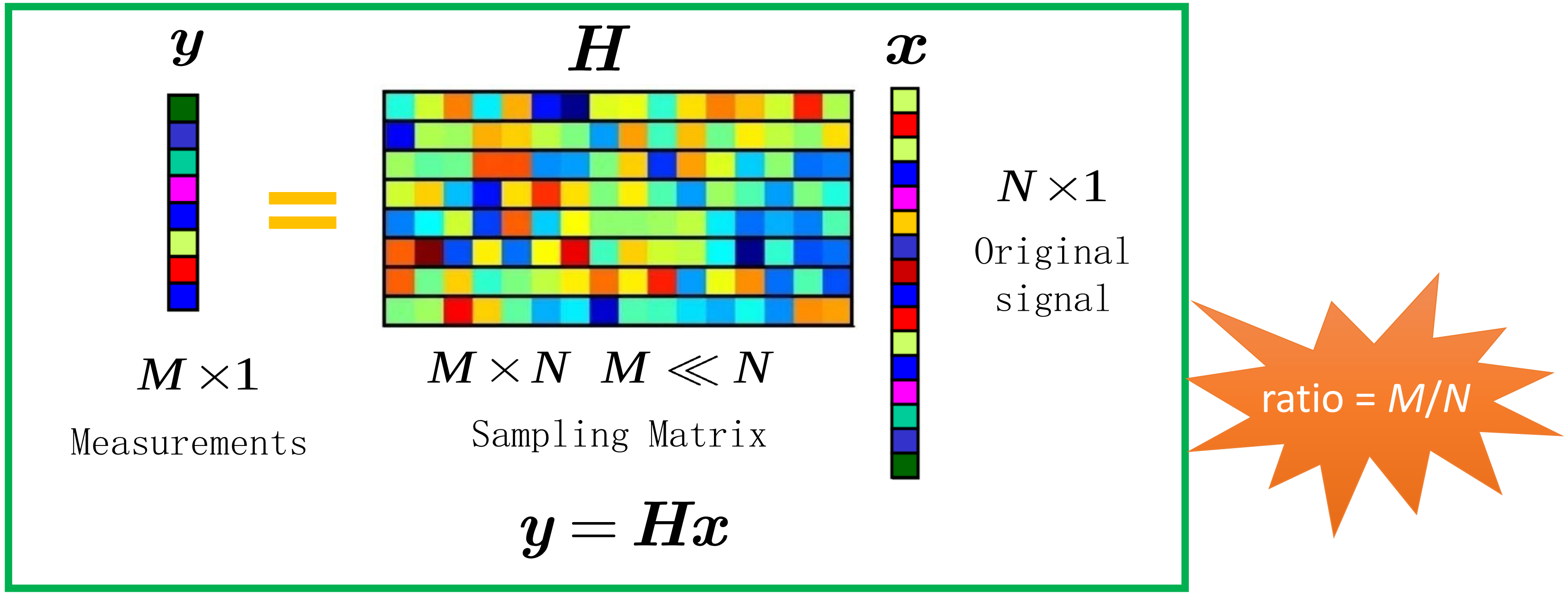


$\otimes$

$\boldsymbol{x}$

Image Reconstruction

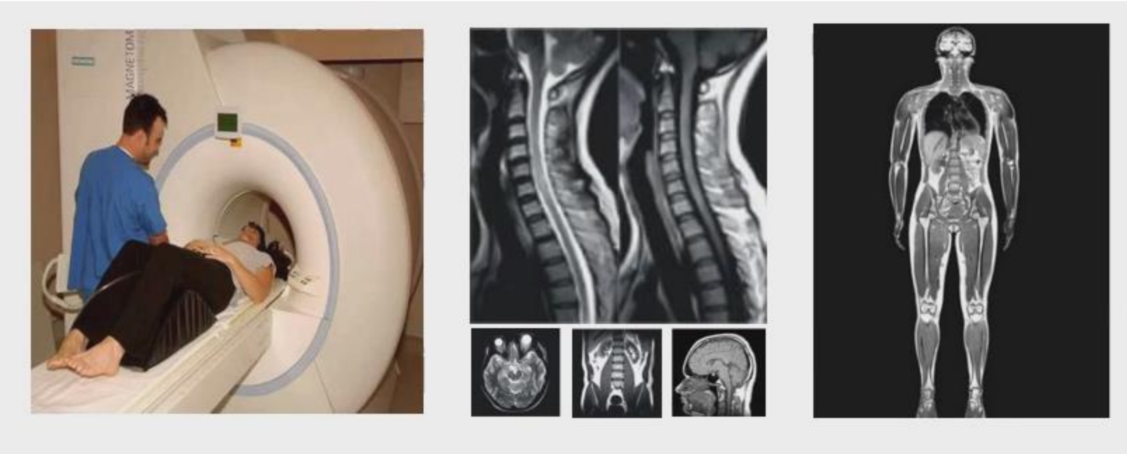
- Compressive Sensing



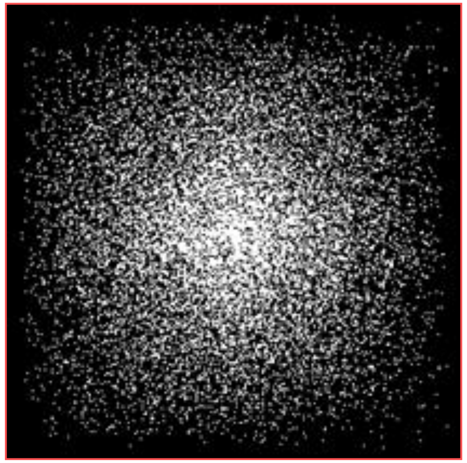
CS theory shows that a signal $\boldsymbol{x}$ can be decoded from many fewer measurements $\boldsymbol{y}$ than suggested by the Nyquist sampling theory, when the signal is sparse in some domain Ψ.

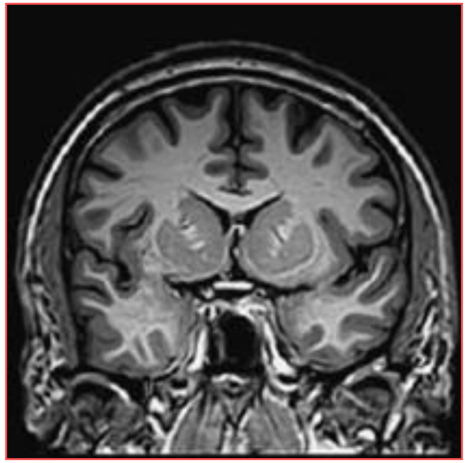
Image Reconstruction

- MRI-CS



$y = Hx + n$





x

$H = PF$ ***P*** is a under-sampling matrix, and ***F*** is a Fourier transform

Image Reconstruction

Observation: $$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n}$$

Regularization-based Framework

Formulation: $$\min_{x} \frac{1}{2}\|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda\,\psi(\boldsymbol{x})$$

Data Fidelity Term

Prior Term

Where

$\boldsymbol{H}$ is a random projection, the problem is image CS recovery;

$\boldsymbol{H}$ is a blur operator, the problem is image deblurring;

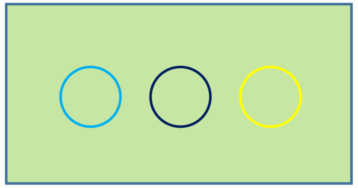
$\boldsymbol{H}$ is a mask operator, the problem is image inpainting;

$\boldsymbol{H}$ is an identity, the problem is image denoising.

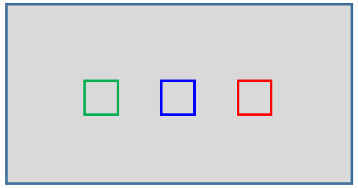
Image Reconstruction

- Image Property

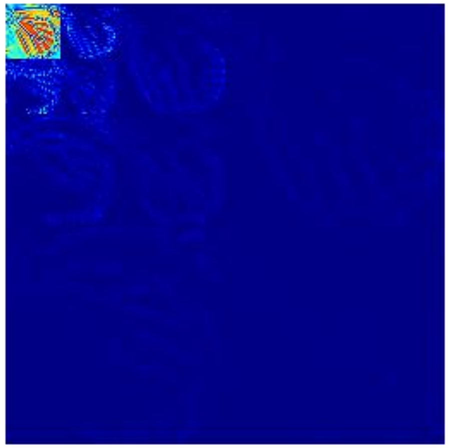
$$\min_{x} \frac{1}{2} \| \boldsymbol{Hx} - \boldsymbol{y} \|_2^2 + \lambda\, \psi(\boldsymbol{x})$$



Smoothness



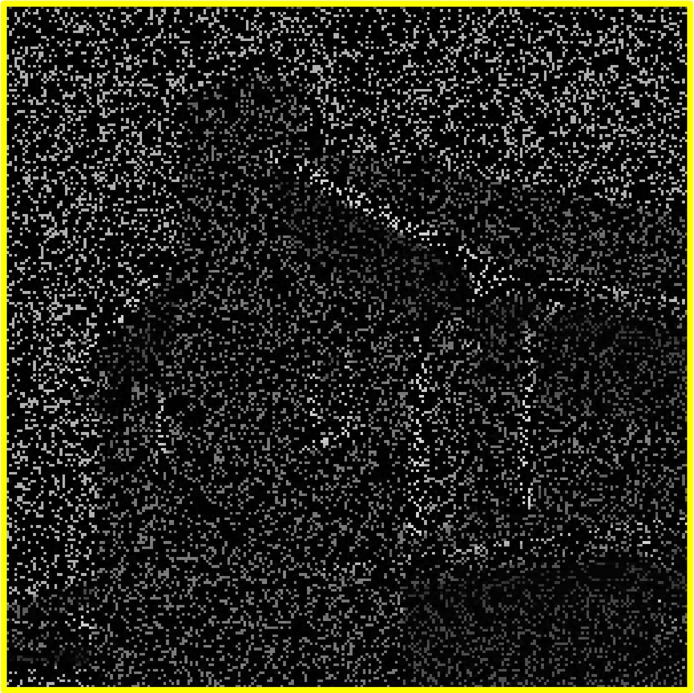
Self-similarity



Sparsity

Image Reconstruction

- $$\min_{x} \frac{1}{2} \| \boldsymbol{Hx} - \boldsymbol{y} \|_2^2 + \lambda\, \psi(\boldsymbol{x})$$



Jian Zhang, et. al. Group-based Sparse Representation for Image Restoration. IEEE Transactions on Image Processing, 2014.

Image Reconstruction

Proximal splitting methods

$$\min_{\mathbf{z}} f_1(\mathbf{z}) + f_2(\mathbf{z})$$

Fixed-step proximal descent

- Initialize: $\mathbf{z}^0$
- Gradient step: $\mathbf{b}^{k+1} = \mathbf{z}^k - \frac{1}{\alpha}\nabla f_1(\mathbf{z}^k)$
- Proximal step: $\mathbf{z}^{k+1} = \mathrm{prox}_{\frac{1}{\alpha}f_2}\left(\mathbf{b}^{k+1}\right)$
- Iterate until convergence...

Proximal operator

$$\mathrm{prox}_{f_2}(\mathbf{z}) = \arg\min_{\mathbf{u}} \|\mathbf{u} - \mathbf{z}\|_2^2 + f_2(\mathbf{u})$$

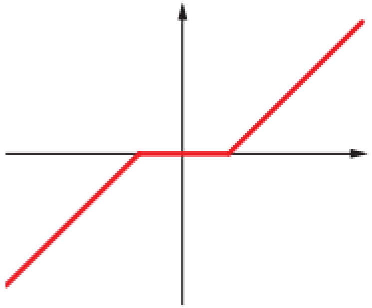
Image Reconstruction

Iterated Shrinkage-Thresholding Algorithm (ISTA)

$$\min_{\mathbf{z}} \frac{1}{2} \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2 + \lambda \|\mathbf{z}\|_1$$

Proximal operator: element-wise soft thresholding

$$\boldsymbol{\pi}_{\boldsymbol{\lambda}} = \max\{0, |\mathbf{z}| - \lambda\}\text{sign}(\mathbf{z})$$



- Initialize: $\mathbf{z}^0 = \mathbf{0}$
- Gradient step: $\mathbf{b}^{k+1} = \frac{1}{\alpha}\mathbf{D}^{\mathrm{T}}\mathbf{x} + (\mathbf{I} - \frac{1}{\alpha}\mathbf{D}^{\mathrm{T}}\mathbf{D})\mathbf{z}^k$
- Proximal step: $\mathbf{z}^{k+1} = \boldsymbol{\pi}_{\frac{\lambda}{\alpha}}\left(\mathbf{b}^{k+1}\right)$
- Iterate until convergence…

Image Reconstruction

$$\min_{x} \frac{1}{2}\|\boldsymbol{Hx} - \boldsymbol{y}\|_2^2 + \lambda\,\psi(\boldsymbol{x})$$

$\boldsymbol{x} = \mathbf{W}a, \mathbf{W}$ is wavelet basis

$$\arg\min_{\mathbf{a}} \|\mathbf{y} - \mathbf{HWa}\|_2^2 + \lambda\|\mathbf{a}\|_1$$

ISTA

$$\mathbf{a}^{k+1} = \operatorname*{prox}_{l_1}(\mathbf{a}^k, \lambda) \quad \triangleq \quad \mathcal{S}_{\lambda/L}\left(\frac{1}{L}\mathbf{W}^*\mathbf{H}^*\mathbf{y} + (\mathbf{I} - \frac{1}{L}\mathbf{W}^*\mathbf{H}^*\mathbf{HW})\mathbf{a}^k\right)$$

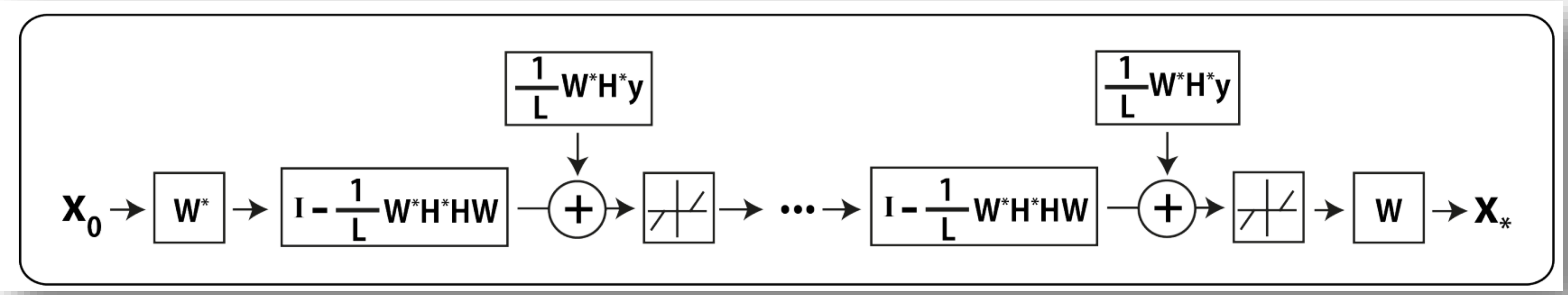
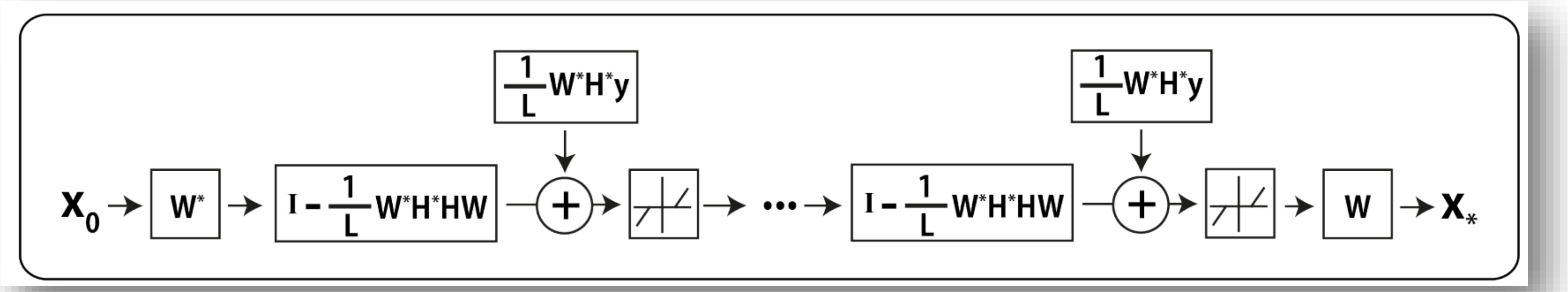
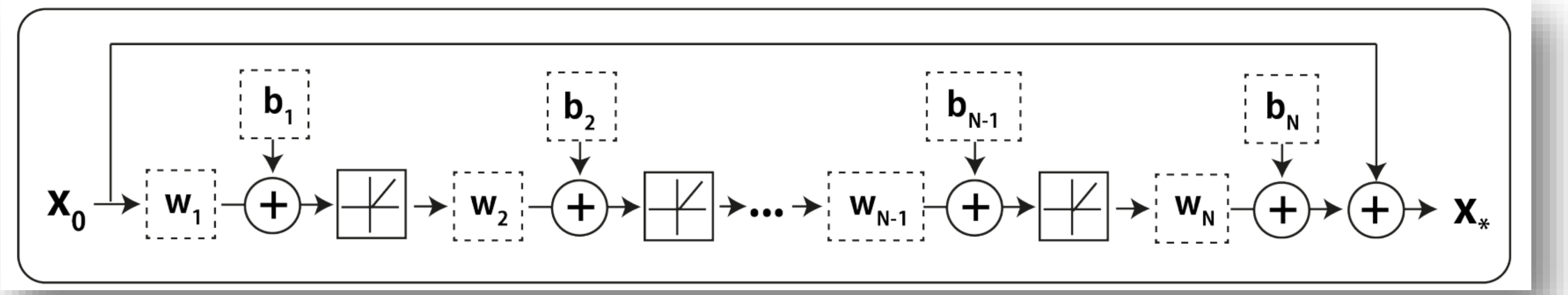


Image Reconstruction

ISTA



CNN/FCN

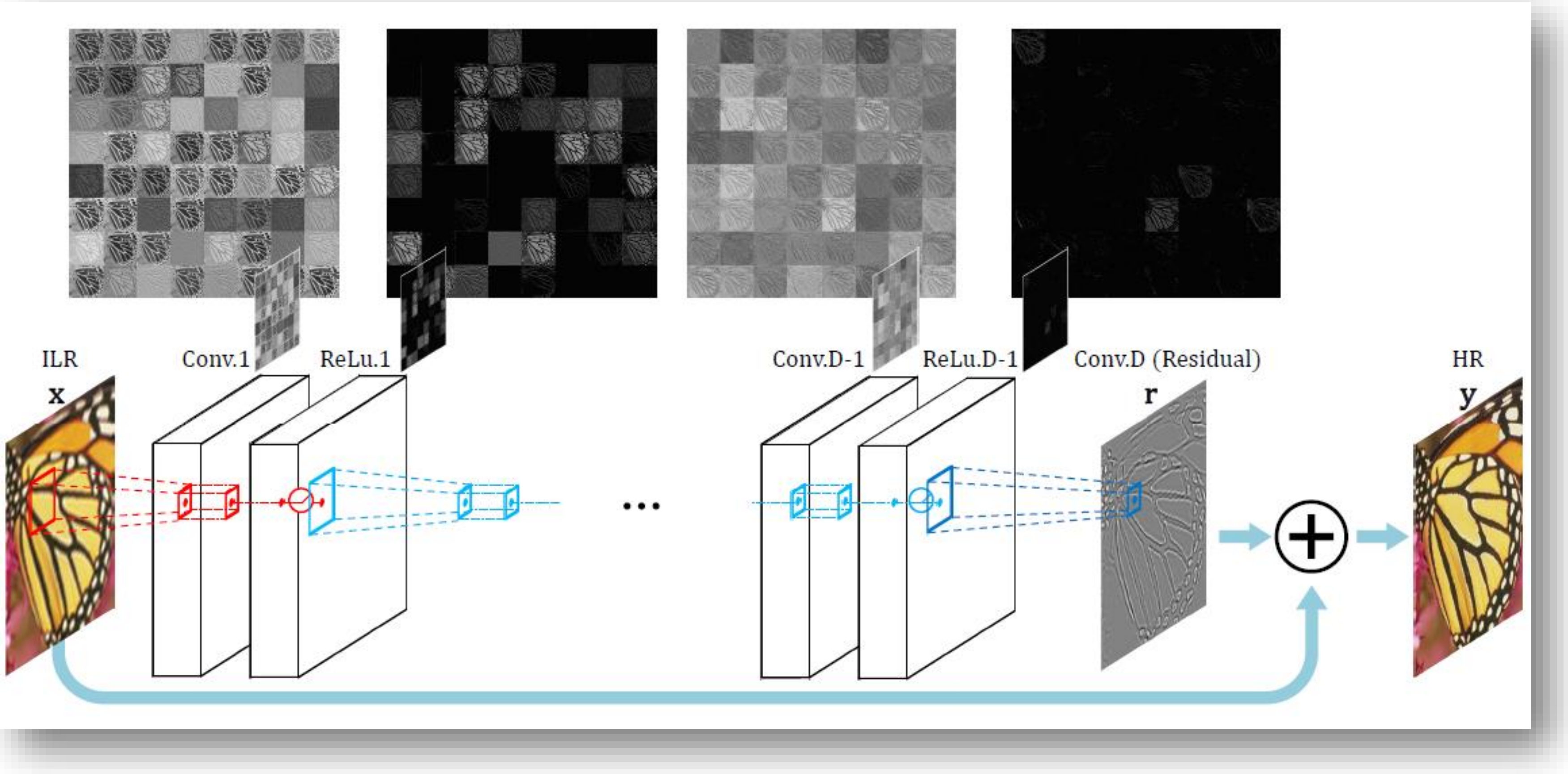


K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2010, pp. 399–406.

Contents

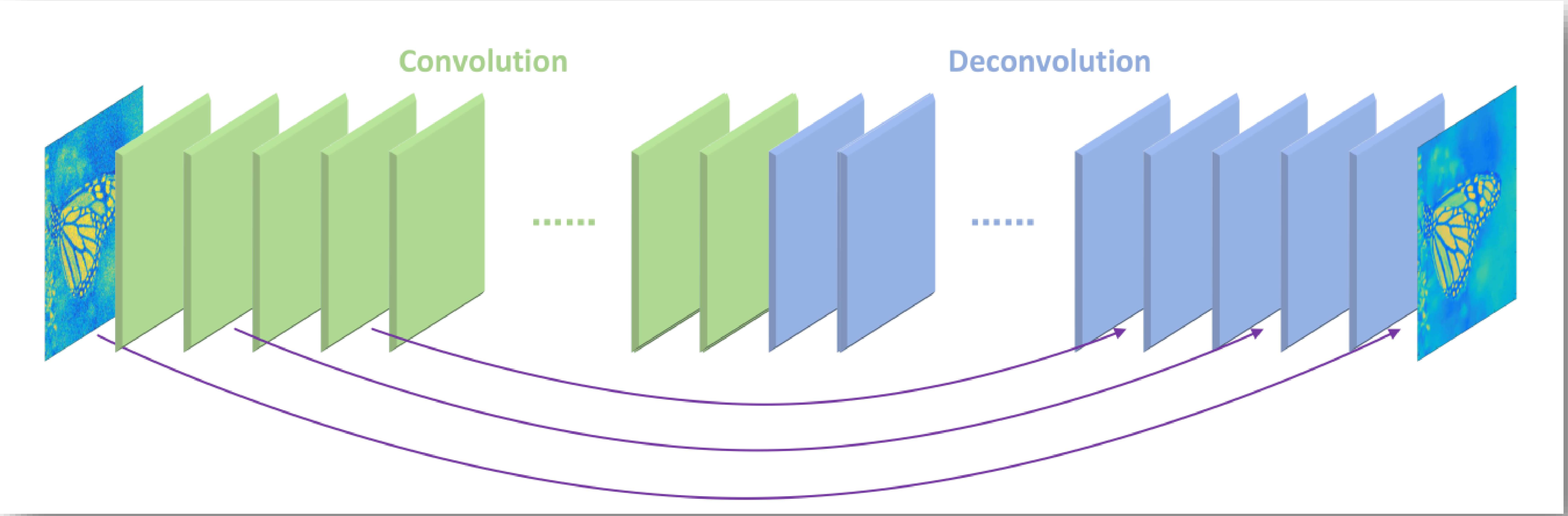
- Regularization-based Framework for Image Reconstruction
  - Denoising, Inpainting, Deblurring, Compressive Sensing
  - Regularization-based Framework
  - Optimization (Proximal Splitting, Iterated Shrinkage-Thresholding)
- Deep Convolutional Neural Network for Image Reconstruction
  - Super-resolution
  - Image Denoising
- **Deep ADMM-Net for Compressive Sensing MRI**

Deep Convolutional Neural Network



J. Kim, J. K. Lee, and K. M. Lee, “Accurate image super-resolution using very deep convolutional networks,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Deep Convolutional Neural Network



X.-J. Mao, C. Shen, and Y.-B. Yang, Image restoration using very deep fully convolutional encoder-decoder networks with symmetric skip connections, *NIPS*, 2016

Problems

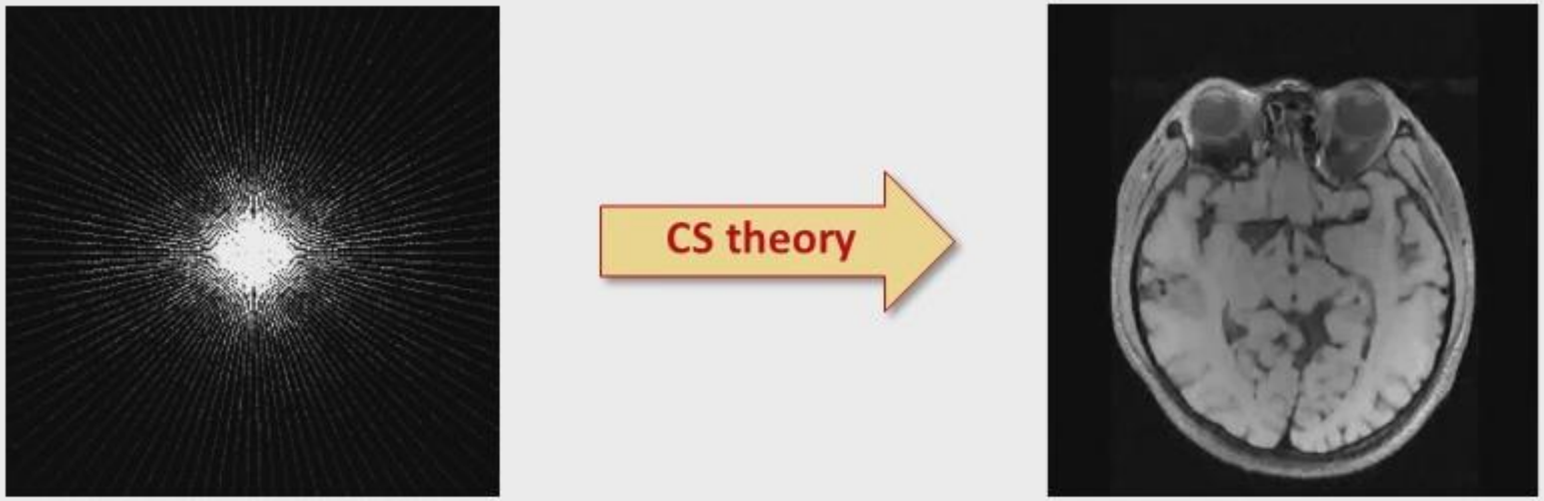
- Image Reconstruction Under Regularization-based Framework
  - Hand-crafted prior term
  - Many tuned parameter
  - Too many iterations
  - Speed and Quality
- Image Reconstruction via Deep Convolutional Neural Network
  - Naive and Simple: end to end learning, many layers (deep)
  - Layer organization may be not efficient

Deep ADMM-Net for Compressive Sensing MRI

- **CS-MRI** is an effective approach for fast Magnetic Resonance Imaging (MRI).

Deep ADMM-Net for Compressive Sensing MRI

- **CS-MRI** is an effective approach for fast Magnetic Resonance Imaging (MRI).



Under-Sampled data in k-space

Reconstructed MR image

Deep ADMM-Net for Compressive Sensing MRI

General CS-MRI Model: Assume $x \in \mathbb{C}^N$ is an MRI image to be reconstructed, $y \in \mathbb{C}^{N'}$ ($N' < N$) is the under-sampled k-space data, according to the CS theory, the reconstructed image can be estimated by solving the following optimization problem:

$$\boxed{\hat{x} = \arg\min_{x} \left\{ \frac{1}{2}\|Ax - y\|_2^2 + \sum_{l=1}^{L} \lambda_l g(D_l x) \right\},} \tag{1}$$

where $A = PF \in \mathbb{R}^{N' \times N}$ is a measurement matrix, $P \in \mathbb{R}^{N' \times N}$ is a under-sampling matrix, and F is a Fourier transform. D_l denotes a transform matrix for a filtering operation, e.g., Discrete Wavelet Transform (DWT), Discrete Cosine Transform (DCT), etc. $g(\cdot)$ is a regularization function derived from the data prior, e.g., l_q-norm ($0 \leq q \leq 1$) for a sparse prior. λ_l is a regularization parameter.

Deep ADMM-Net for Compressive Sensing MRI

ADMM solver: [12] The above optimization problem can be solved efficiently using ADMM algorithm. By introducing auxiliary variables $z = \{z_1, z_2, \cdots, z_L\}$, Eqn. (1) is equivalent to:

$$\min_{x,z} \frac{1}{2}\|Ax - y\|_2^2 + \sum_{l=1}^{L} \lambda_l g(z_l) \qquad s.t. \;\; z_l = D_l x, \;\; \forall\, l \in [1, 2, \cdots, L]. \tag{2}$$

Its augmented Lagrangian function is :

$$\mathfrak{L}_\rho(x, z, \alpha) = \frac{1}{2}\|Ax - y\|_2^2 + \sum_{l=1}^{L} \lambda_l g(z_l) - \sum_{l=1}^{L} \langle \alpha_l, z_l - D_l x \rangle + \sum_{l=1}^{L} \frac{\rho_l}{2}\|z_l - D_l x\|_2^2, \tag{3}$$

where $\alpha = \{\alpha_l\}$ are Lagrangian multipliers and $\rho = \{\rho_l\}$ are penalty parameters. ADMM alternatively optimizes $\{x, z, \alpha\}$ by solving the following three subproblems:

Deep ADMM-Net for Compressive Sensing MRI

$$\begin{cases} x^{(n+1)} = \arg\min_x \frac{1}{2}\|Ax-y\|_2^2 - \sum_{l=1}^{L}\langle \alpha_l^{(n)}, z_l^{(n)} - D_l x\rangle + \sum_{l=1}^{L} \frac{\rho_l}{2}\|z_l^{(n)} - D_l x\|_2^2, \\ z^{(n+1)} = \arg\min_z \sum_{l=1}^{L} \lambda_l g(z_l) - \sum_{l=1}^{L}\langle \alpha_l^{(n)}, z_l - D_l x^{(n+1)}\rangle + \sum_{l=1}^{L} \frac{\rho_l}{2}\|z_l - D_l x^{(n+1)}\|_2^2, \\ \alpha^{(n+1)} = \arg\min_\alpha \sum_{l=1}^{L}\langle \alpha_l, D_l x^{(n+1)} - z_l^{(n+1)}\rangle, \end{cases}$$

where $n \in [1, 2, \cdots, N_s]$ denotes n-th iteration. For simplicity, let $\beta_l = \frac{\alpha_l}{\rho_l}$ $(l \in [1, 2, \cdots, L])$, and substitute $A = PF$ into Eqn. (4). Then the three subproblems have the following solutions:

Deep ADMM-Net for Compressive Sensing MRI

$$
\begin{cases}
\mathbf{X^{(n)}} : x^{(n)} = F^T[P^T P + \sum_{l=1}^{L} \rho_l F D_l^T D_l F^T]^{-1}[P^T y + \sum_{l=1}^{L} \rho_l F D_l^T (z_l^{(n-1)} - \beta_l^{(n-1)})], \\
\mathbf{Z^{(n)}} : z_l^{(n)} = S(D_l x^{(n)} + \beta_l^{(n-1)}; \lambda_l/\rho_l), \\
\mathbf{M^{(n)}} : \beta_l^{(n)} = \beta_l^{(n-1)} + \eta_l (D_l x^{(n)} - z_l^{(n)}),
\end{cases}
\tag{5}
$$

where $x^{(n)}$ can be efficiently computed by fast Fourier transform, $S(\cdot)$ is a nonlinear shrinkage function. It is usually a soft or hard thresholding function corresponding to the sparse regularization of l_1-norm and l_0-norm respectively [20]. The parameter η_l is an update rate.

Deep ADMM-Net for Compressive Sensing MRI

- **General CS-MRI Model**

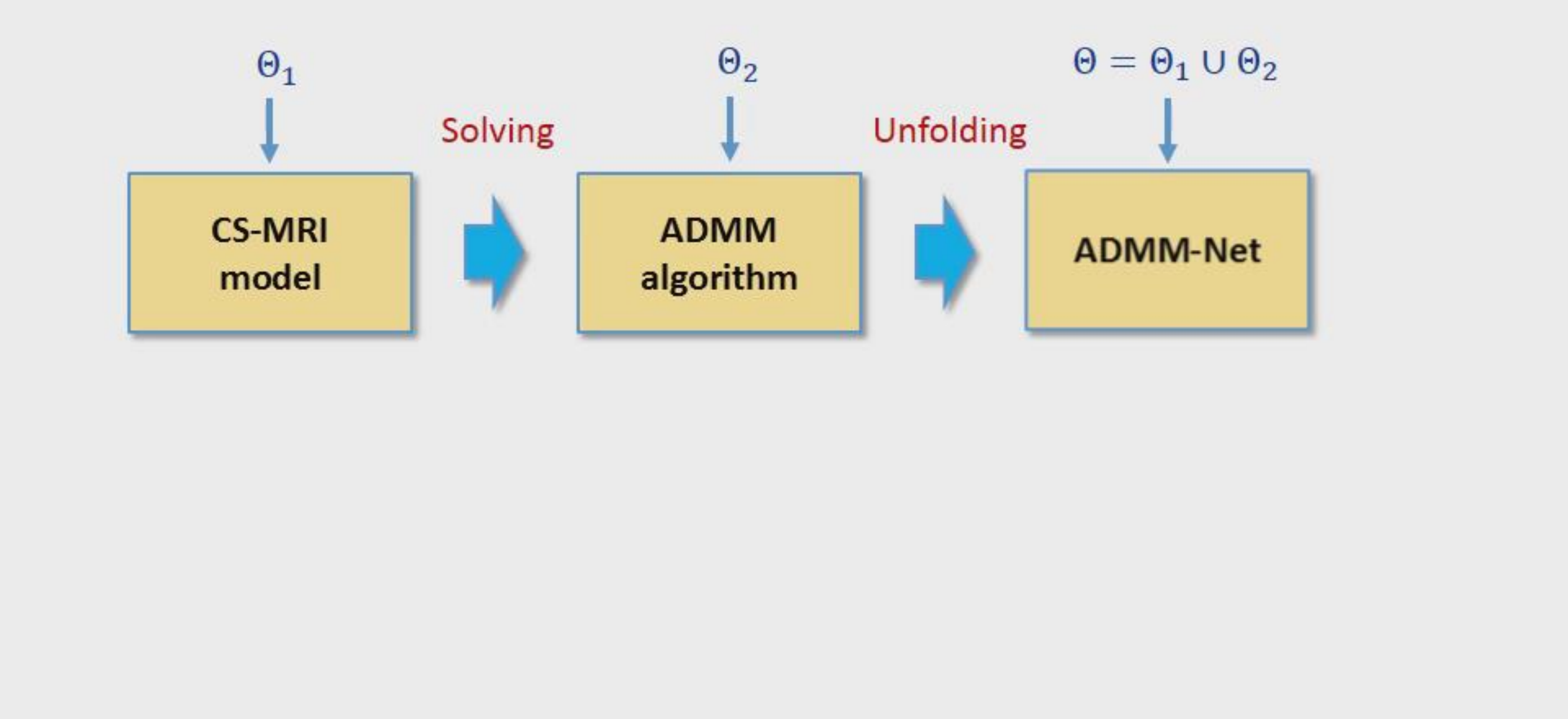
$$\hat{x} = \arg\min_{x}\left\{\frac{1}{2}\|Ax-y\|_2^2 + \sum_{l=1}^{L}\lambda_l g(D_l x)\right\} \qquad A = PF$$

- **ADMM algorithm**

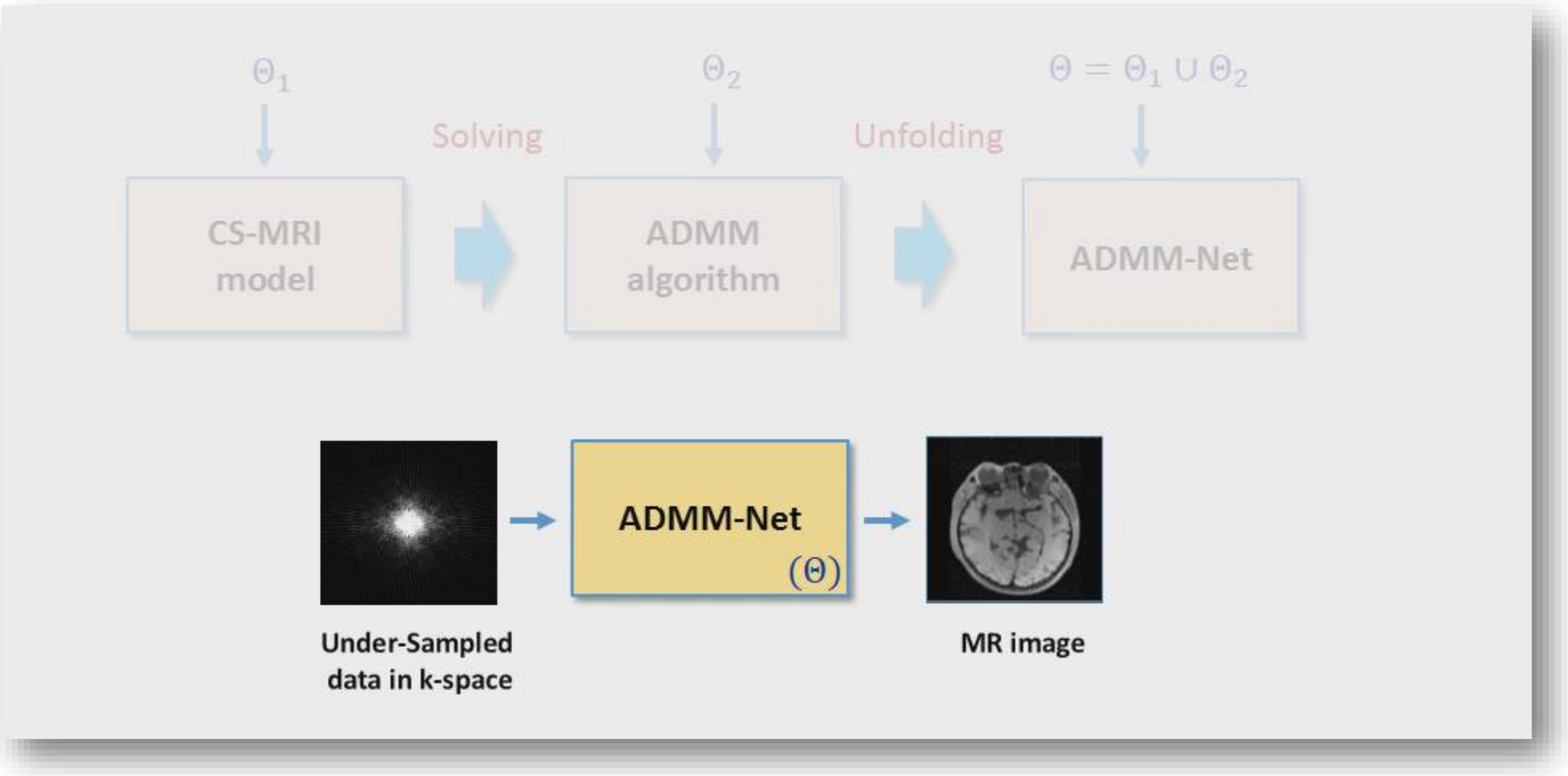
$$\begin{cases} \boldsymbol{X}^{(n)}:\ \boldsymbol{x}^{(n)} = \arg\min_{\boldsymbol{x}} L_\rho\left(\boldsymbol{x}, \boldsymbol{z}^{(n-1)}, \boldsymbol{\beta}^{(n-1)}\right) = \boldsymbol{F}^T\left(\boldsymbol{P}^T\boldsymbol{P} + \sum_l \rho_l \boldsymbol{F}\boldsymbol{D}_l^T\boldsymbol{D}_l\boldsymbol{F}^T\right)^{-1}\left[\boldsymbol{P}^T\boldsymbol{y} + \sum_l \rho_l \boldsymbol{F}\boldsymbol{D}_l^T\left(\boldsymbol{z}_l^{(n-1)} - \boldsymbol{\beta}_l^{(n-1)}\right)\right] \\ \boldsymbol{Z}^{(n)}:\ \boldsymbol{z}_l^{(n)} = \arg\min_{\boldsymbol{z}_l} L_\rho\left(\boldsymbol{x}^{(n)}, \boldsymbol{z}_l, \boldsymbol{\beta}^{(n-1)}\right) = S\left(\boldsymbol{D}_l\boldsymbol{x}^{(n)} + \boldsymbol{\beta}_l^{(n-1)}; \lambda_l/\rho_l\right) \\ \boldsymbol{M}^{(n)}:\ \boldsymbol{\beta}_l^{(n)} = \boldsymbol{\beta}_l^{(n-1)} + \eta_l\left(\boldsymbol{D}_l\boldsymbol{x}^{(n)} - \boldsymbol{z}_l^{(n)}\right) \end{cases}$$

Challenges!

Deep ADMM-Net for Compressive Sensing MRI



Deep ADMM-Net for Compressive Sensing MRI



Deep ADMM-Net for Compressive Sensing MRI

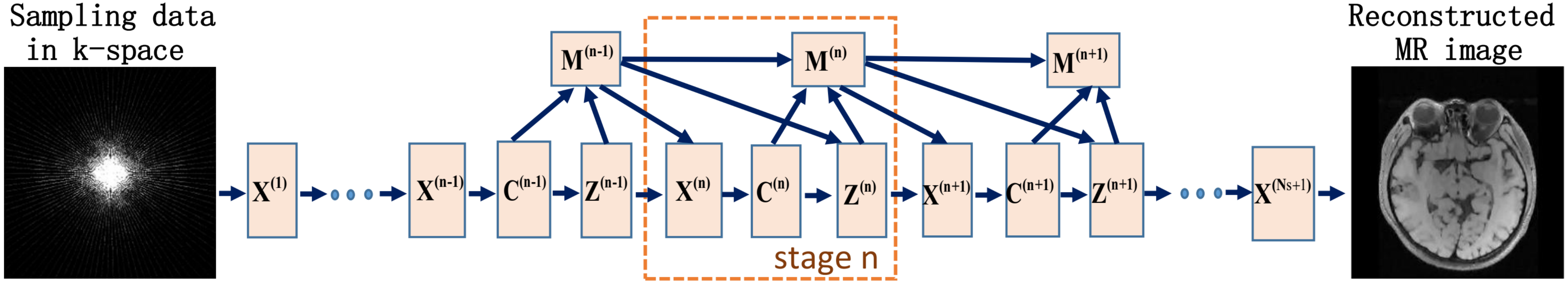
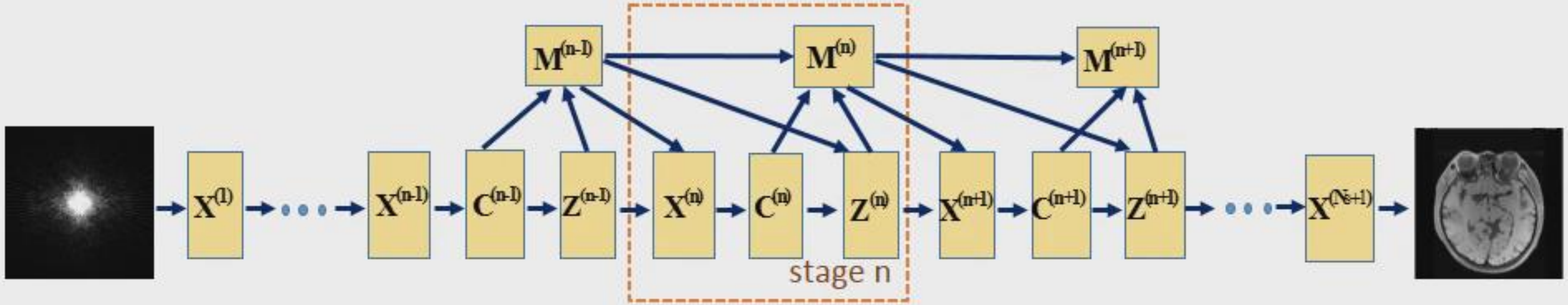


Figure 1: The data flow graph for the ADMM optimization of a general CS-MRI model. This graph consists of four types of nodes: reconstruction ($\mathbf{X}$), convolution ($\mathbf{C}$), non-linear transform ($\mathbf{Z}$), and multiplier update ($\mathbf{M}$). An under-sampled data in k-space is successively processed over the graph, and finally generates a MR image. Our deep ADMM-Net is defined over this data flow graph.

Deep ADMM-Net for Compressive Sensing MRI

- **Network structure**



ADMM Iterations:

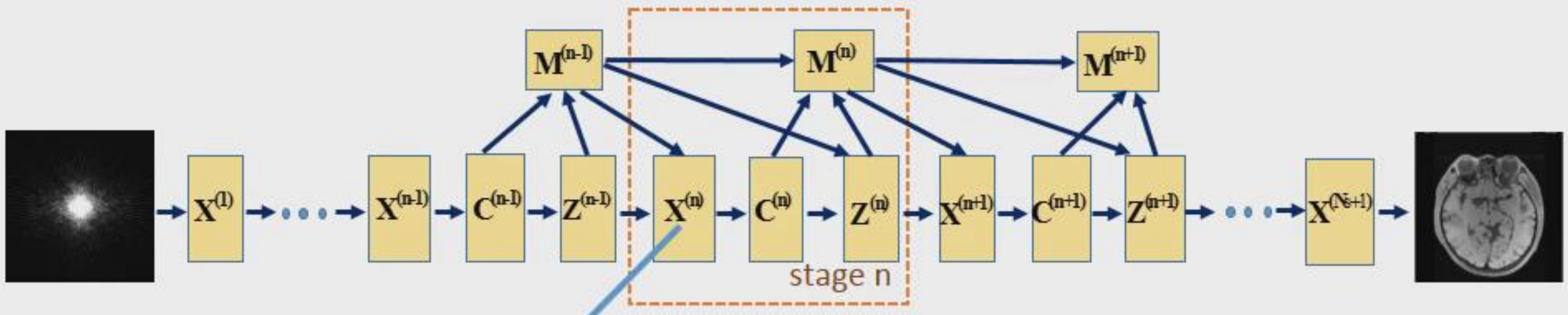
$$X^{(n)}:\ x^{(n)} = F^T\left(P^TP + \sum_l \rho_l F D_l^T D_l F^T\right)^{-1}\left[P^T y + \sum_l \rho_l F D_l^T\left(z_l^{(n-1)} - \beta_l^{(n-1)}\right)\right]$$

$$Z^{(n)}:\ z_l^{(n)} = S\left(D_l x^{(n)} + \beta_l^{(n-1)}; \lambda_l/\rho_l\right)$$

$$M^{(n)}:\ \beta_l^{(n)} = \beta_l^{(n-1)} + \eta_l\left(D_l x^{(n)} - z_l^{(n)}\right)$$

Deep ADMM-Net for Compressive Sensing MRI

- **Reconstruction layer($X^{(n)}$)**



ADMM Iterations:

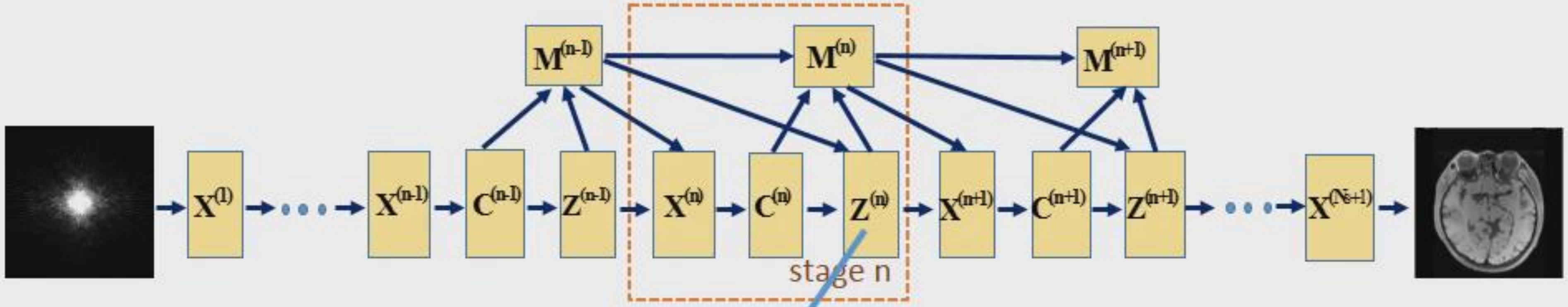
$$\boxed{X^{(n)}:\ x^{(n)} = F^T\left(P^TP + \sum_l \rho_l F D_l^T D_l F^T\right)^{-1}\left[P^Ty + \sum_l \rho_l F D_l^T\left(z_l^{(n-1)} - \beta_l^{(n-1)}\right)\right]}$$

$$Z^{(n)}:\ z_l^{(n)} = S\left(D_l x^{(n)} + \beta_l^{(n-1)};\ {}^{\lambda_l}/_{\rho_l}\right)$$

$$M^{(n)}:\ \beta_l^{(n)} = \beta_l^{(n-1)} + \eta_l\left(D_l x^{(n)} - z_l^{(n)}\right)$$

Deep ADMM-Net for Compressive Sensing MRI

- **Non-linear transform layer($Z^{(n)}$):**

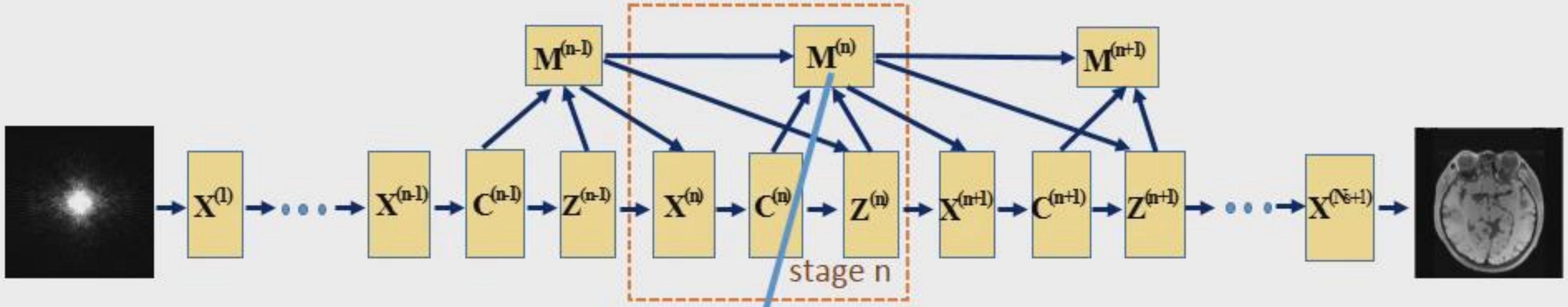


ADMM Iterations:

$$\begin{cases} X^{(n)}:\ x^{(n)} = F^T\left(P^TP + \sum_l \rho_l F D_l^T D_l F^T\right)^{-1}\left[P^Ty + \sum_l \rho_l F D_l^T\left(z_l^{(n-1)} - \beta_l^{(n-1)}\right)\right] \\ \boxed{Z^{(n)}:\ z_l^{(n)} = S\left(D_l x^{(n)} + \beta_l^{(n-1)}; \lambda_l/\rho_l\right)} \\ M^{(n)}:\ \beta_l^{(n)} = \beta_l^{(n-1)} + \eta_l\left(D_l x^{(n)} - z_l^{(n)}\right) \end{cases}$$

Deep ADMM-Net for Compressive Sensing MRI

- **Multiplier update layer($M^{(n)}$):**



ADMM Iterations:

$$X^{(n)}:\ x^{(n)} = F^T\left(P^TP + \sum_l \rho_l F D_l^T D_l F^T\right)^{-1}\left[P^T y + \sum_l \rho_l F D_l^T\left(z_l^{(n-1)} - \beta_l^{(n-1)}\right)\right]$$

$$Z^{(n)}:\ z_l^{(n)} = S\left(D_l x^{(n)} + \beta_l^{(n-1)}; \lambda_l/\rho_l\right)$$

$$\boxed{M^{(n)}:\ \beta_l^{(n)} = \beta_l^{(n-1)} + \eta_l\left(D_l x^{(n)} - z_l^{(n)}\right)}$$

Deep ADMM-Net for Compressive Sensing MRI

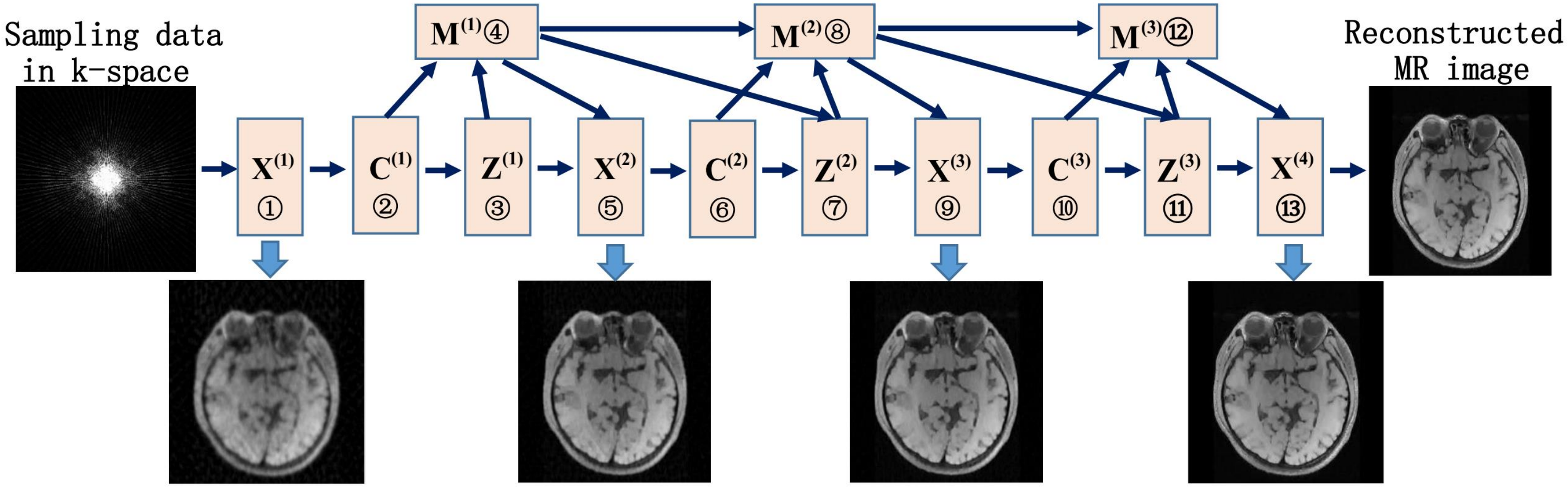


Figure 3: An example of deep ADMM-Net with three stages. The sampled data in k-space is successively processed by operations in a order from 1 to 12, followed by a reconstruction layer $X^{(4)}$ to output the final reconstructed image. The reconstructed image in each stage is shown under each reconstruction layer.

Deep ADMM-Net for Compressive Sensing MRI

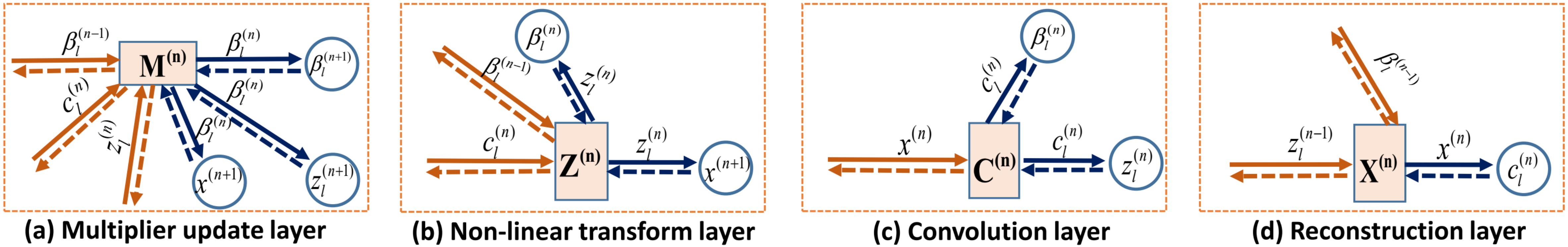
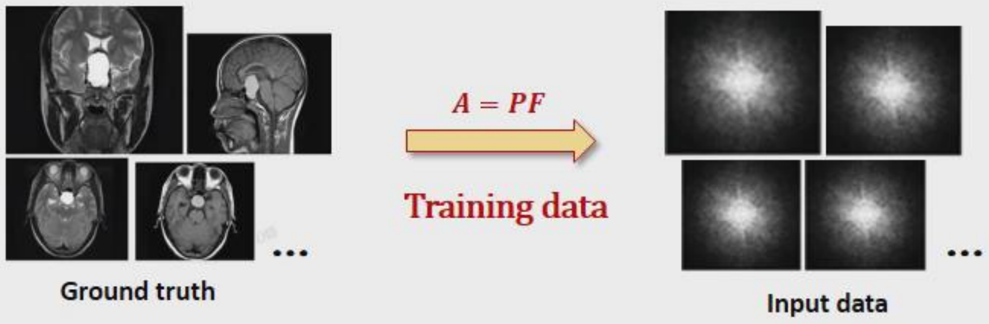


Figure 4: Illustration of four types of graph nodes (i.e., layers in network) and their data flows in stage n. The solid arrow indicates the data flow in forward pass and dashed arrow indicates the backward pass when computing gradients in backpropagation.

Deep ADMM-Net for Compressive Sensing MRI

- **Network training**



Loss function(NMSE):

$$E(\Theta) = \frac{1}{|\Gamma|} \sum_{(y, x^{gt}) \in \Gamma} \frac{\|\hat{x}(y, \Theta) - x^{gt}\|_2}{\|x^{gt}\|_2}$$

Network Parameters: These layers are organized in a data flow graph shown in Fig. 1. In the deep architecture, we aim to learn the following parameters: $H_l^{(n)}$ and $\rho_l^{(n)}$ in reconstruction layer, filters $D_l^{(n)}$ in convolution layer, $\{q_{l,i}^{(n)}\}_{i=1}^{N_c}$ in nonlinear transform layer, $\eta_l^{(n)}$ in multiplier update layer, where $l \in [1, 2, \cdots, L]$ and $n \in [1, 2, \cdots, N_s]$ are the indexes for the filters and stages respectively. All of these parameters are taken as the network parameters to be learned.

Deep ADMM-Net for Compressive Sensing MRI

Table 1: Performance comparisons on brain data with different sampling ratios.

Method	20%		30%		40%		50%		Test time
	NMSE	PSNR	NMSE	PSNR	NMSE	PSNR	NMSE	PSNR	
Zero-filling	0.1700	29.96	0.1247	32.59	0.0968	34.76	0.0770	36.73	0.0013s
TV [2]	0.0929	35.20	0.0673	37.99	0.0534	40.00	0.0440	41.69	0.7391s
RecPF [4]	0.0917	35.32	0.0668	38.06	0.0533	40.03	0.0440	41.71	0.3105s
SIDWT	0.0885	35.66	0.0620	38.72	0.0484	40.88	0.0393	42.67	7.8637s
PBDW [6]	0.0814	36.34	0.0627	38.64	0.0518	40.31	0.0437	41.81	35.3637s
PANO [10]	0.0800	36.52	0.0592	39.13	0.0477	41.01	0.0390	42.76	53.4776s
FDLCP [8]	0.0759	36.95	0.0592	39.13	0.0500	40.62	0.0428	42.00	52.2220s
BM3D-MRI [11]	0.0674	37.98	0.0515	40.33	0.0426	41.99	0.0359	43.47	40.9114s
Init-Net_{13}	0.1394	31.58	0.1225	32.71	0.1128	33.44	0.1066	33.95	0.6914s
ADMM-Net_{13}	0.0752	37.01	0.0553	39.70	0.0456	41.37	0.0395	42.62	0.6964s
ADMM-Net_{14}	0.0742	37.13	0.0548	39.78	0.0448	41.54	0.0380	42.99	0.7400s
ADMM-Net_{15}	0.0739	37.17	0.0544	39.84	0.0447	41.56	0.0379	43.00	0.7911s

Deep ADMM-Net for Compressive Sensing MRI

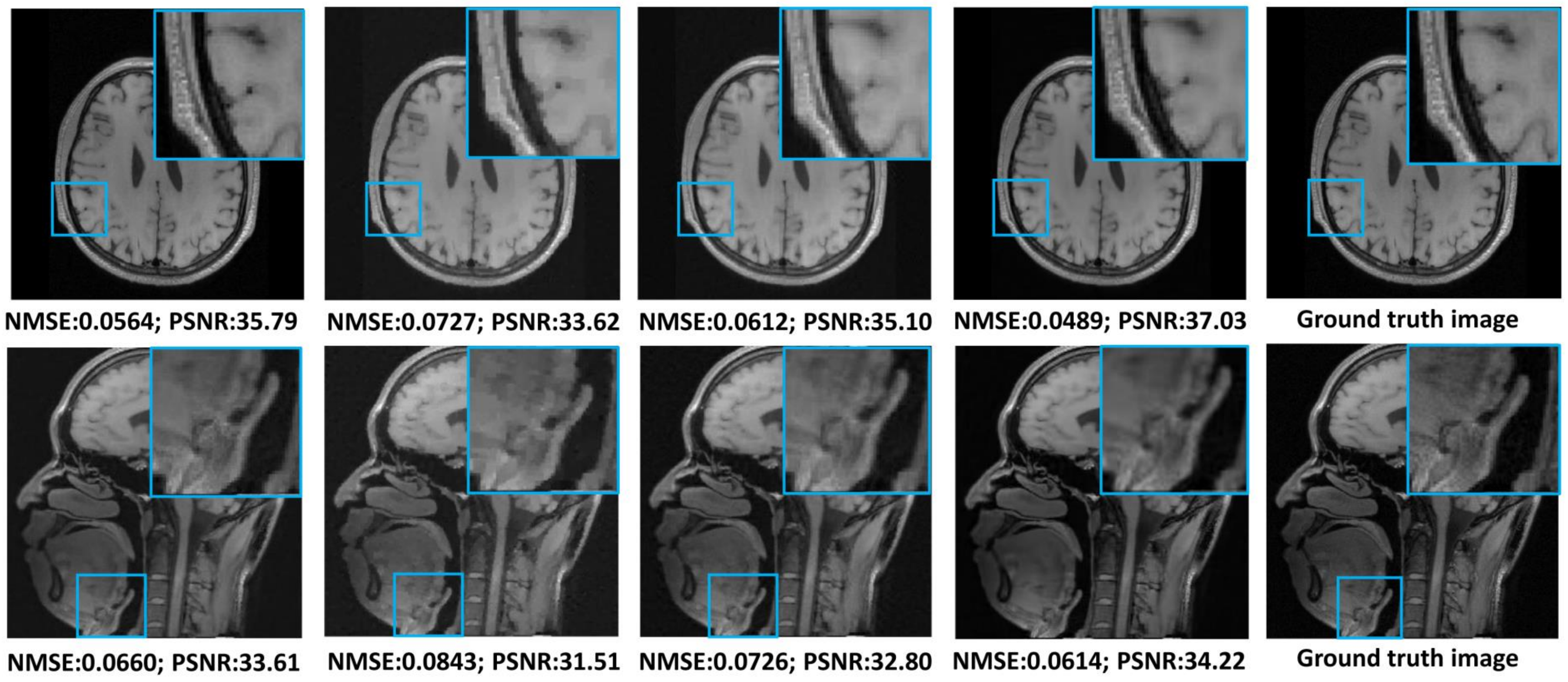


Figure 5: Examples of reconstruction results with 20% (the first row) and 30% (the second row) sampling ratios. The left four columns show results of ADMM-Net$_{15}$, RecPF, PANO, BM3D-MRI.

Deep ADMM-Net for Compressive Sensing MRI

Test time and Effect of the number of stages:

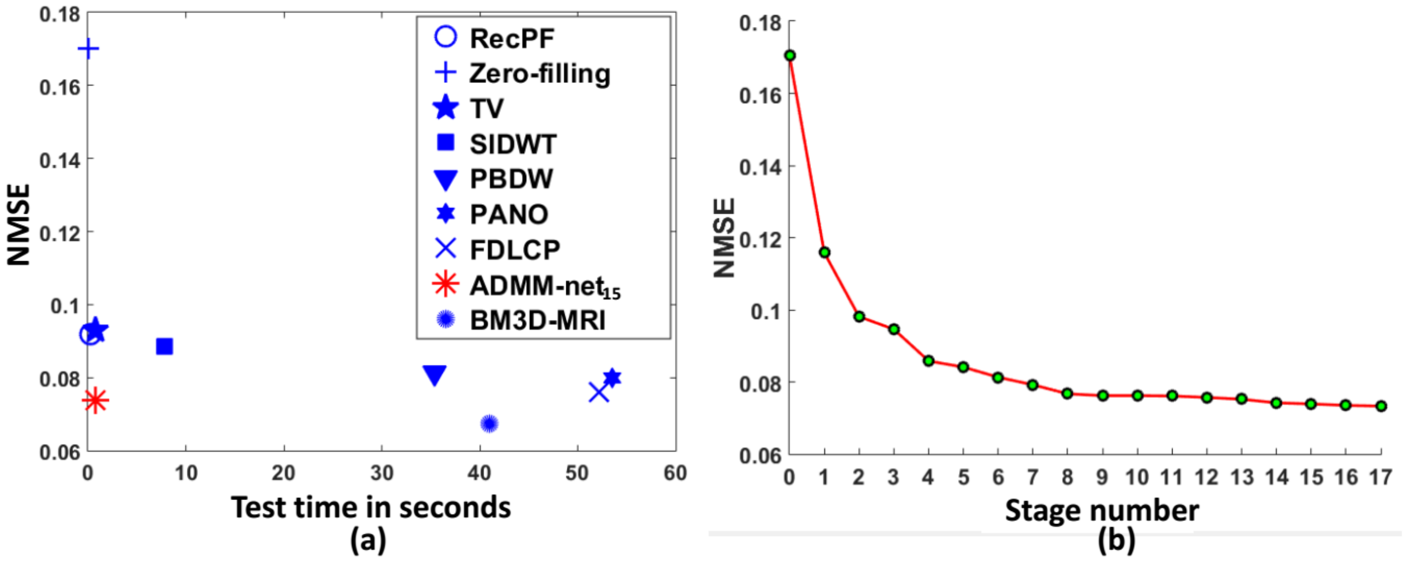


Figure 6: (a) Scatter plot of NMSEs and average test time for different methods; (b) The NMSEs of ADMM-Net using different number of stages (20% sampling ratio for brain data).

Deep ADMM-Net for Compressive Sensing MRI

Effect of the filter size:

Effect of the filter sizes: We also train ADMM-Net initialized by two gradient filters with size of 1×3 and 3×1 respectively for all convolution and reconstruction layers, the corresponding trained net with 13 stages under 20% sampling ratio achieves NMSE value of 0.0899 and PSNR value of 36.52 db on brain data, compared with 0.0752 and 37.01 db using eight 3×3 filters as shown in Tab. I. We also learn ADMM-Net$_{13}$ with 8 filters sized 5×5 initialized by DCT basis, the performance is not significantly improved, but the training and testing time are significantly longer.

Deep ADMM-Net for Compressive Sensing MRI

- Examples of the learned nonlinear functions and the filters

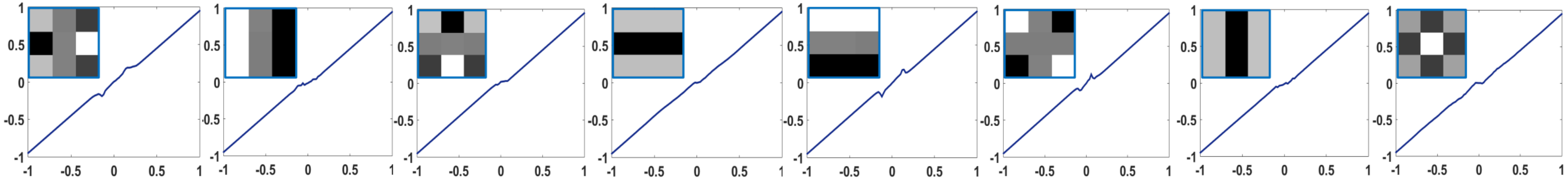


Figure 7: Examples of learned filters in convolution layer and the corresponding nonlinear transforms (the first stage of ADMM-Net_{15} with 20% sampling ratio for brain data).

Conclusions

- Novel deep architecture defined over a data flow graph determined by an ADMM algorithm
- High reconstruction accuracy while keeping the computational efficiency of the ADMM algorithm
- Potentially applied to other applications
- Checking … …
- The relationship between Optimization and Neural Network
- Design More Efficient and Effective Network