Deep Learning and Inverse Problems

21–25 January 2019

KTH Royal Institute of Technology, Stockholm, Sweden

Overview

Deep Learning and Inverse Problems 2019 (DLIP2019) is a one week workshop for researchers and practitioners working on deep learning techniques for inverse problems. Our objective is to enable open discussions on both practical and theoretical aspects, and give researchers time to discuss these problems in depth.

The workshop will feature some invited talks, but we hope that most attendants will also contribute with their own knowledge.

Inverse Problems

$$y = Ax + e$$

Where:

- x = image/model/parameters to estimate,
- y = measurement data,
- A = forward operator, and
- e = additive noise (often Gaussian).

Typically **ill-posed**: no guarantees of existence, uniqueness, or stability of x given y.

Need to incorporate **prior information** on x.

Deep Learning approaches

- Post-processing
  - Deep direct estimation, deep posterior sampling (Adler and Öktem)
  - Deep artifact removal (Hauptmann et al.)
  - Continued SVD (Schwab et al.)

- Learned iterative
  - Learned gradient descent, learned primal-dual (Adler and Öktem)
  - Recurrent inference machines (RIM) (Lønning et al.)

- Learned regularizers
  - Network Tikhonov (NETT) (Haltmeier et al.)
  - Adversarial regularizers (Lunz et al.)
  - Generative network parametrization (Mosser et al.)
  - Deep image priors (Otero Baguer et al.)

Post-processing

Want to find approximate inverse mapping

$$\mathcal{F}_\theta^\dagger(y) \approx x_{true}$$

Use some fixed reconstruction operator $\mathcal{A}^\dagger$ and combine with neural network

$$\mathcal{F}_\theta^\dagger = \mathcal{G}_\theta \circ \mathcal{A}^\dagger$$

Optimize mean squared error on supervised training set.

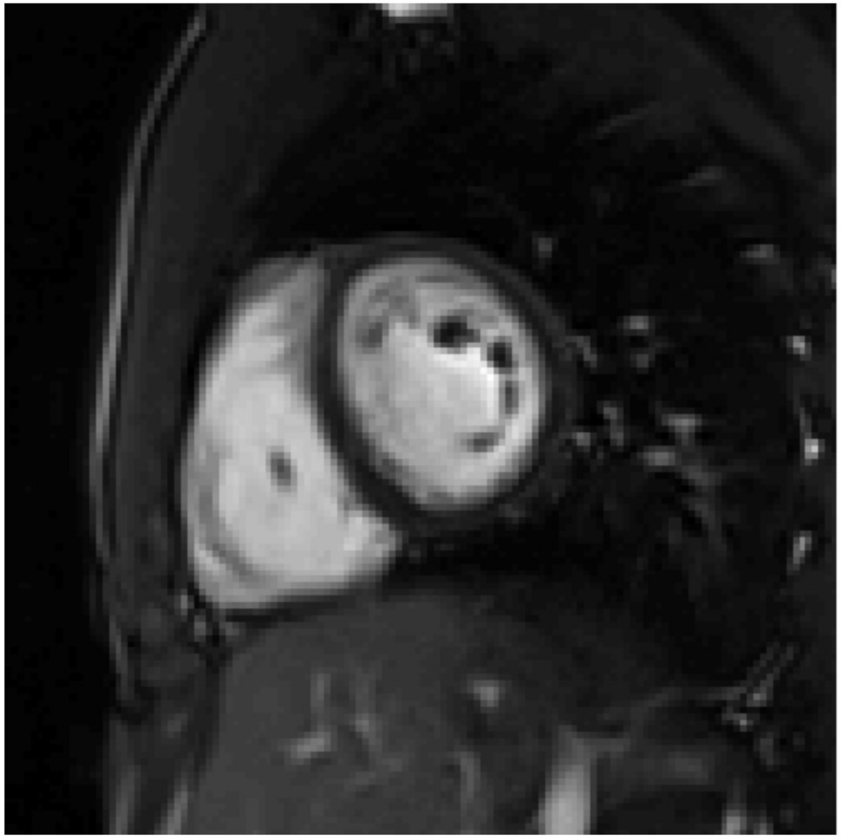
Cardiovascular MR

Forward model: Fourier transform $\mathcal{F}_k$

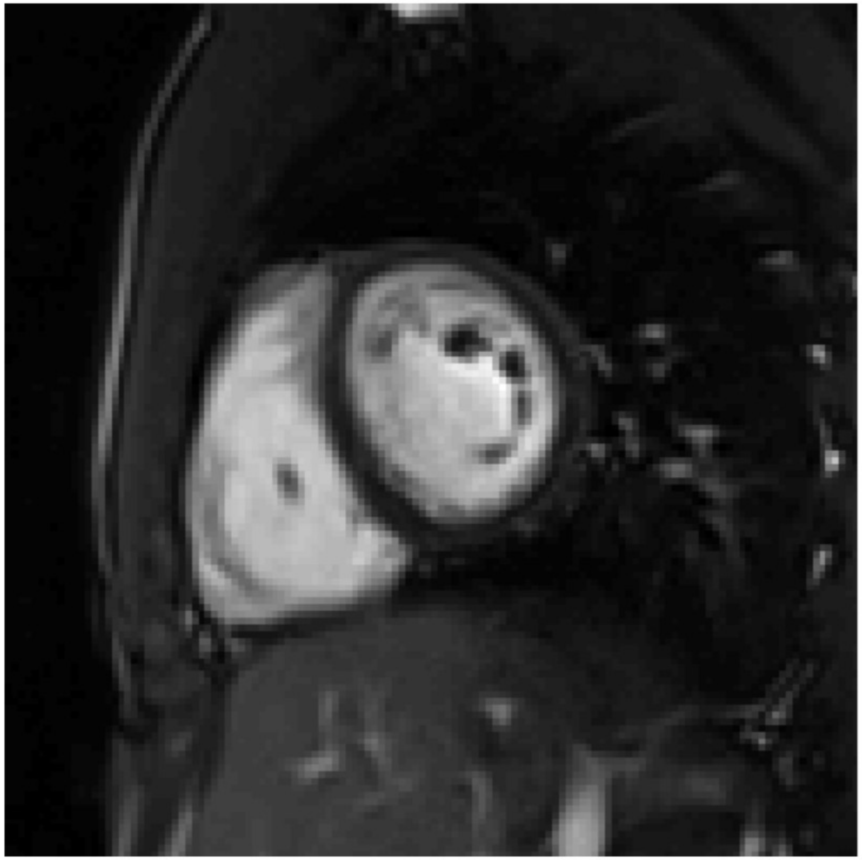
Data given in k-space: $y = \mathcal{F}_k x$

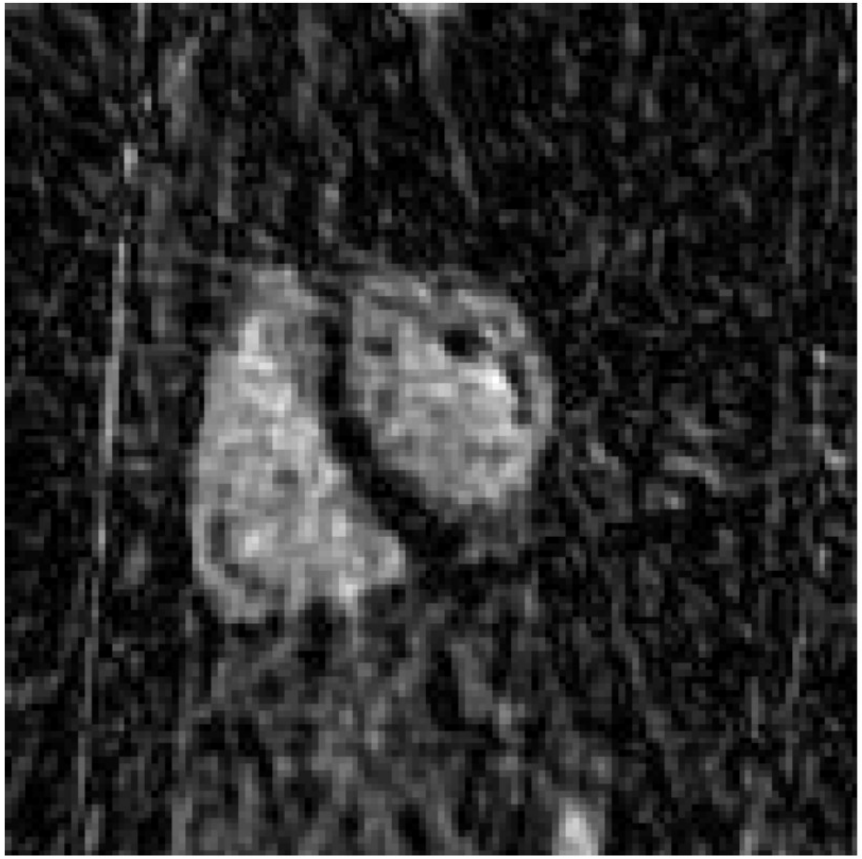
Reconstruction by inverse Fourier transform: $\mathcal{F}_k^{-1} y$

Gold-standard breath-hold

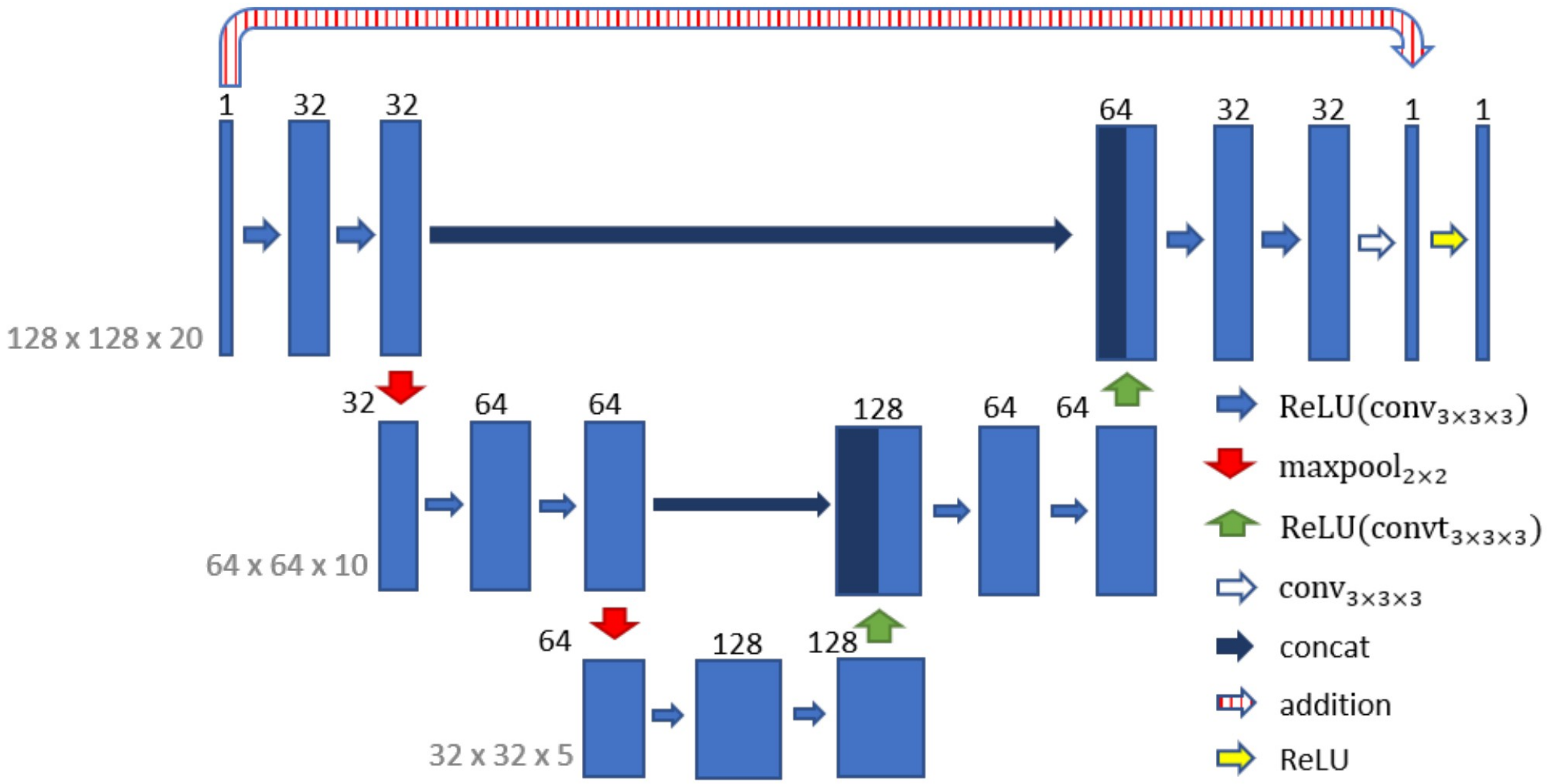


Undersampled reconstruction (13x)





U-Net reconstruction

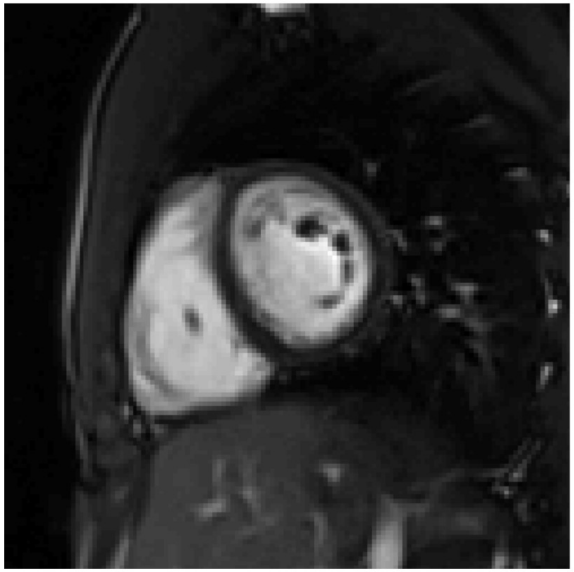


The network $\mathcal{G}_\theta$ was trained to minimise the ℓ^2-distance:

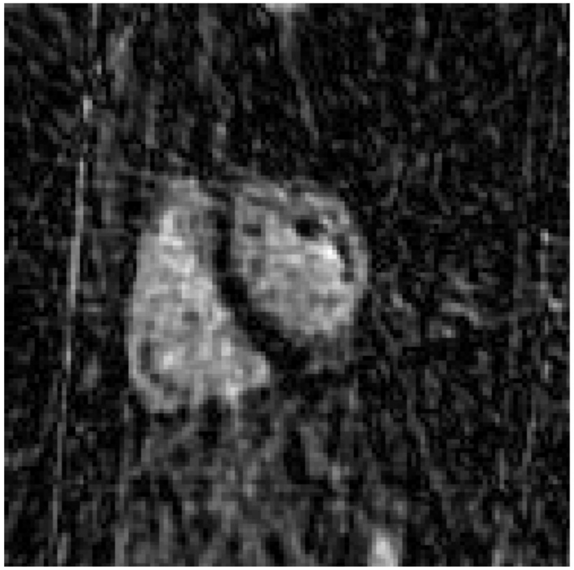
$$\text{loss}(\theta; y^i) = \|\mathcal{G}_\theta(\mathcal{F}_k^{-1} y^i) - x^i_{true}\|_2^2.$$

Comparison

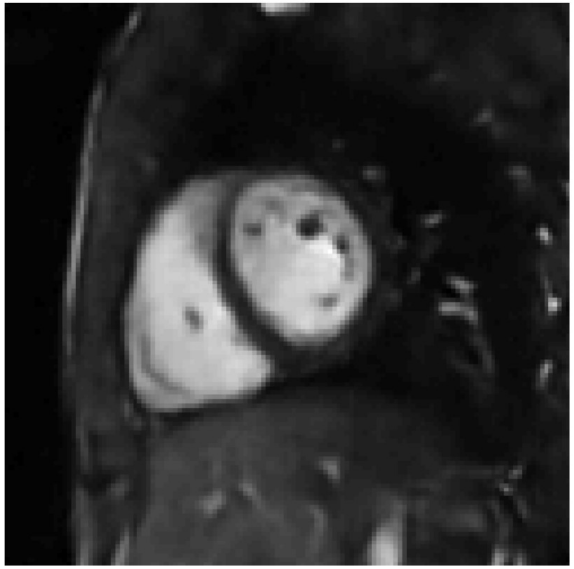
Gold-standard breath-hold



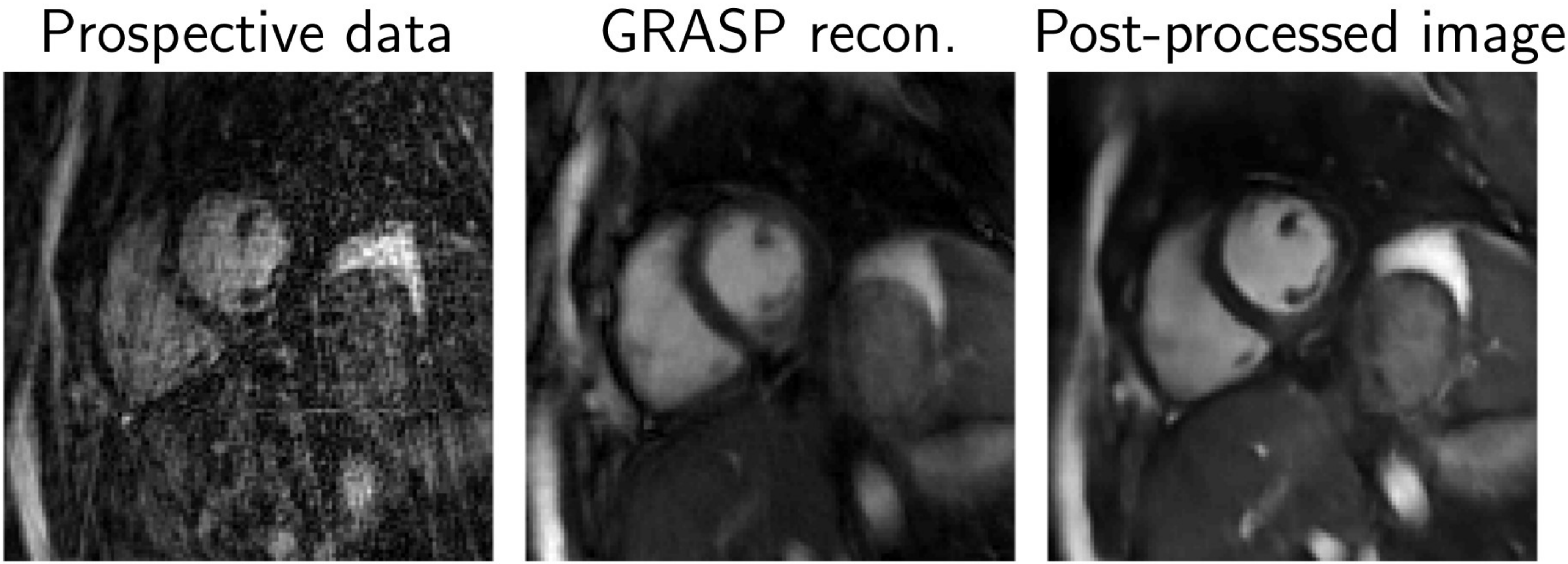
Undersampled reconstruction



Post-processed image



Comparison (cont.)



Learned iterative

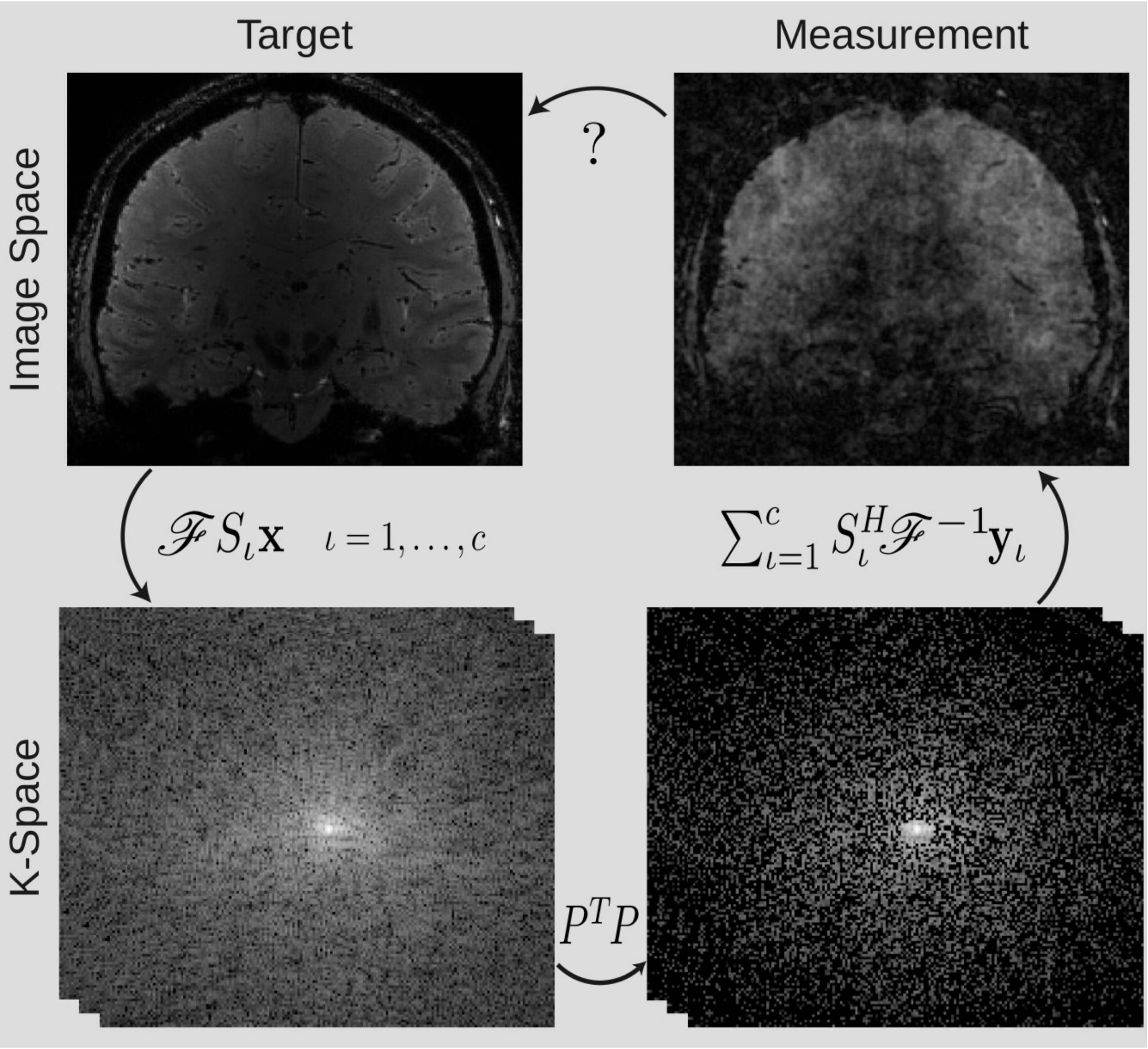
Parametrize update rule

$$x_{k+1} = \mathcal{G}_\theta(\nabla d(y, Ax_k), x_k)$$

where d measures data fit (e.g., log-likelihood).

Stop at K iterations and optimize mean squared error of x_K on supervised dataset.

Accelerated MR imaging



Recurrent inference machine

Parametrize iterative update step

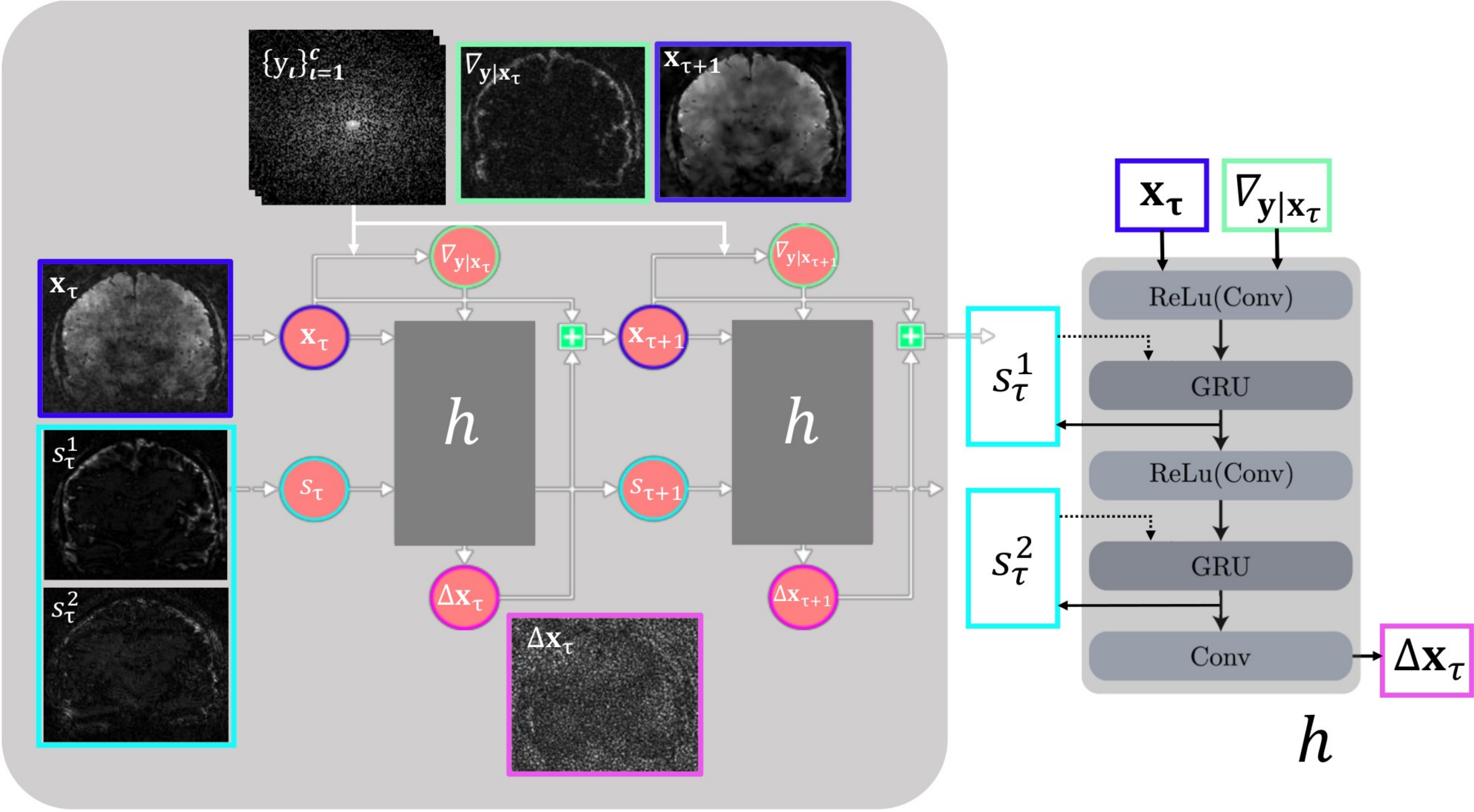
$$\mathbf{s}_0 = \mathbf{0}, \qquad \mathbf{x}_0 = \sum_{\iota=1}^{c} S_\iota^H \mathscr{F}^{-1} P^T \mathbf{y}_\iota,$$

$$\boldsymbol{s}_{\tau+1} = g\left(\nabla_{\mathbf{y}|\mathbf{x}_\tau}, \mathbf{x}_\tau, \mathbf{s}_\tau\right), \quad \mathbf{x}_{\tau+1} = \mathbf{x}_\tau + h\left(\nabla_{\mathbf{y}|\mathbf{x}_\tau}, \mathbf{x}_\tau, \mathbf{s}_{\tau+1}\right),$$

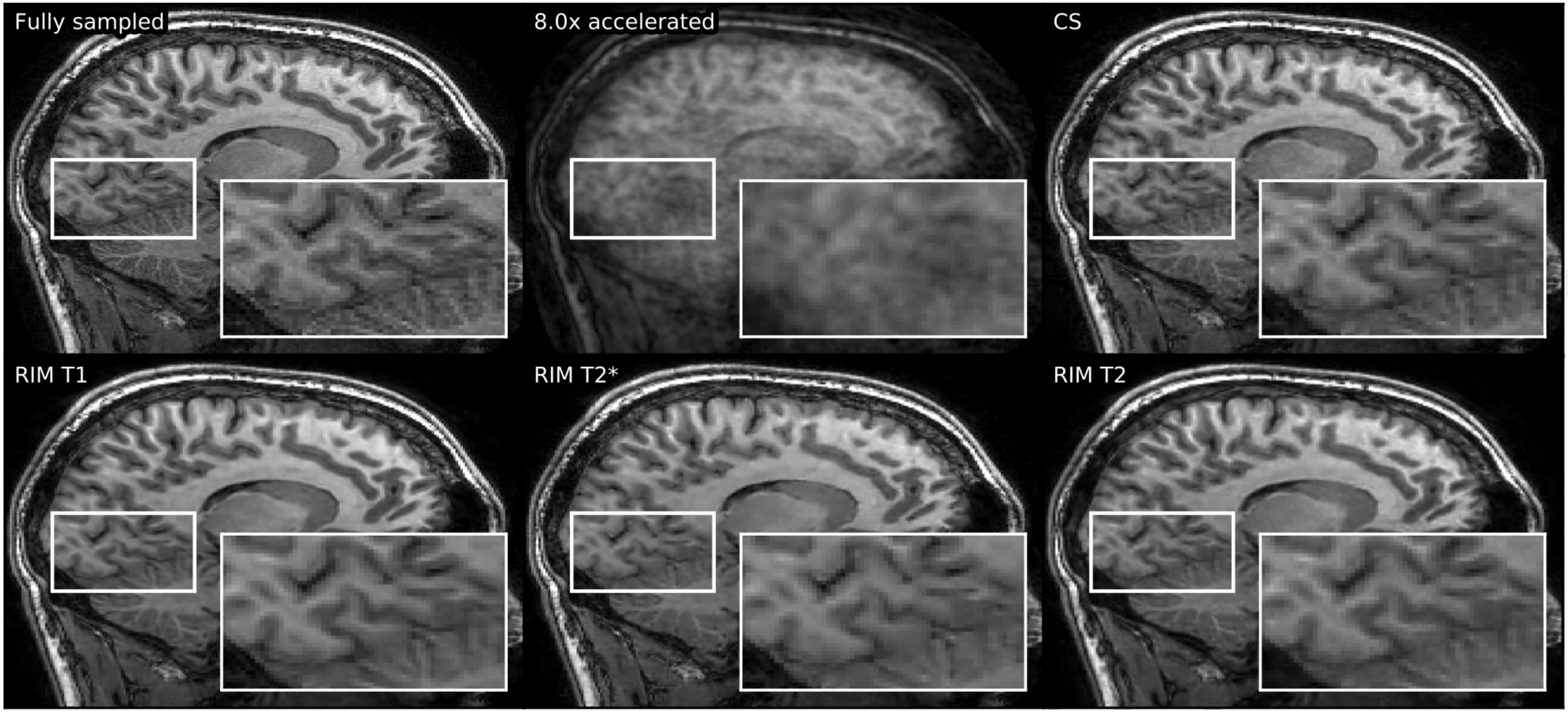
Optimize with weighted mean squared error loss

$$L(\mathbf{x}_t) = \frac{1}{nt}\sum_{\tau=1}^{t} w_\tau \left\|\mathbf{x}_\tau - \mathbf{x}\right\|_2^2 .$$

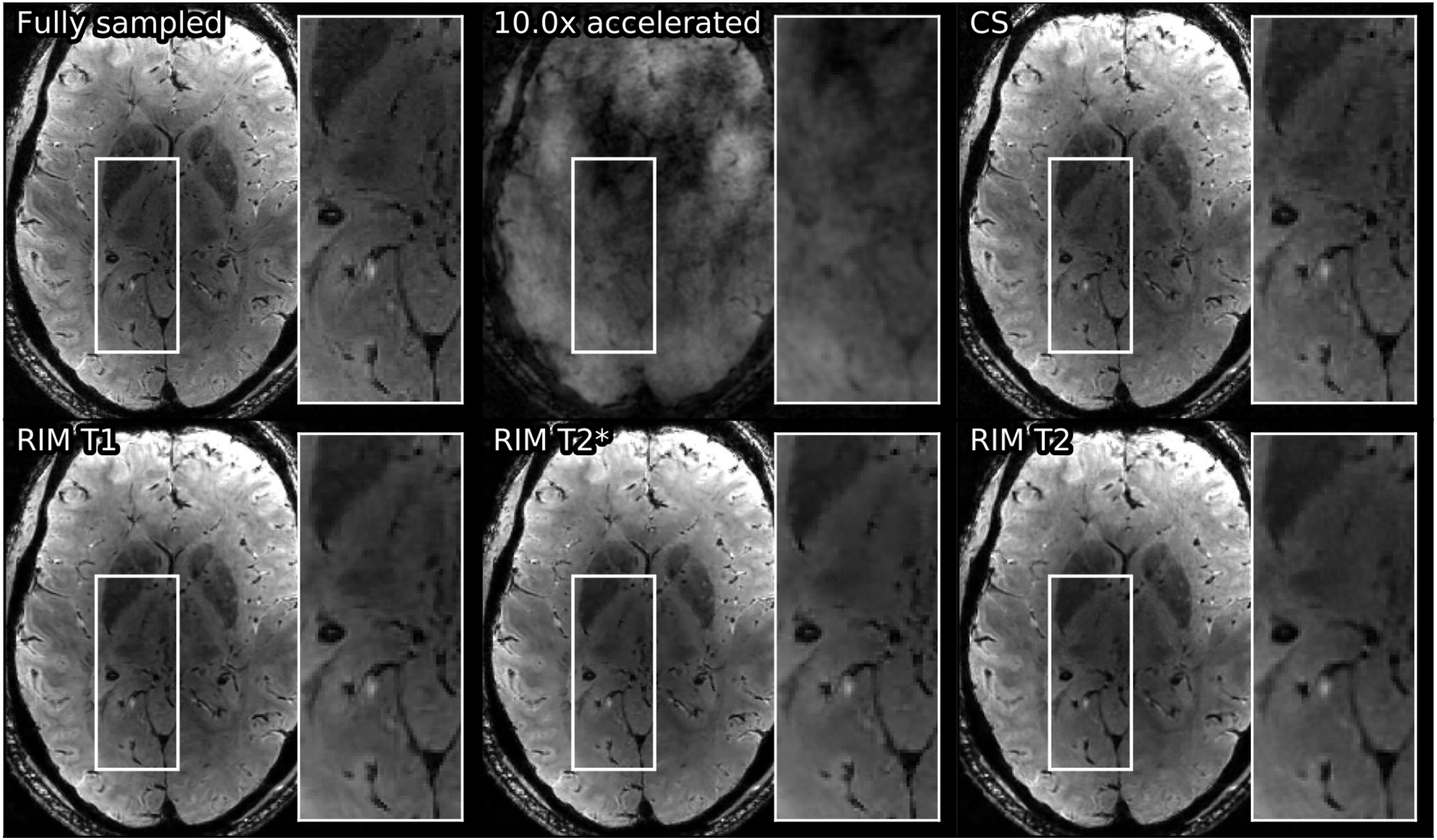
Recurrent inference machine (cont.)



Reconstruction



Reconstuction (cont.)



Learned regularizer

Minimize regularized loss

$$T(x) = d(Ax, y) + \lambda |G_\theta(x)|$$

where regularizer is defined using a neural network.

Optimize network to assign low amplitudes to “real” images and high amplitudes to others on unsupervised data.

S. Lunz

Adversarial regularizers

Distribution of true images $\mathbb{P}_r$

Distibution of (noisy) measurements $\mathbb{P}_Y$

Distribution of “noisy” reconstructions

$$\mathbb{P}_n = (A_\delta^\dagger)_\# \mathbb{P}_Y$$

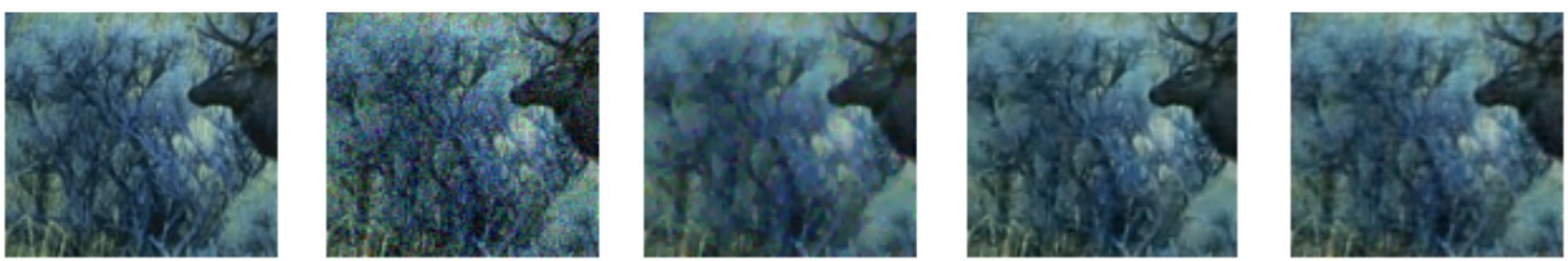
Train Wasserstein GAN to discriminate

$$\mathrm{Wass}(\mathbb{P}_r, \mathbb{P}_n) = \sup_{f \in 1-Lip} \mathbb{E}_{X \sim \mathbb{P}_n}[f(X)] - \mathbb{E}_{X \sim \mathbb{P}_r}[f(X)]$$

Denoising results

Table 1: Denoising results on BSDS dataset

Method	PSNR (dB)	SSIM
Noisy Image	20.3	.534
MODEL-BASED		
Total Variation [23]	26.3	.836
SUPERVISED		
Denoising N.N. [28]	28.8	.908
UNSUPERVISED		
Adversarial Regularizer (ours)	28.2	.892



(a) Ground Truth (b) Noisy Image (c) TV (d) Denoising N.N. (e) Adversarial Reg.

Figure 1: Denoising Results on BSDS

Tomography results

Table 2: CT reconstruction on LIDC dataset

(a) High noise

Method	PSNR (dB)	SSIM
MODEL-BASED		
Filtered Backprojection	14.9	.227
Total Variation [18]	27.7	.890
SUPERVISED		
Post-Processing [15]	31.2	.936
RED [21]	29.9	.904
UNSUPERVISED		
Adversarial Reg. (ours)	30.5	.927

(b) Low noise

Method	PSNR (dB)	SSIM
MODEL-BASED		
Filtered Backprojection	23.3	.604
Total Variation [18]	30.0	.924
SUPERVISED		
Post-Processing [15]	33.6	.955
RED [21]	32.8	.947
UNSUPERVISED		
Adversarial Reg. (ours)	32.5	.946

(a) Ground Truth (b) FBP (c) TV (d) Post-Processing (e) Adversarial Reg.

Figure 2: Reconstruction from simulated CT measurements on the LIDC dataset