Trends in mathematics of information: Deep learning, artificial intelligence and compressed sensing

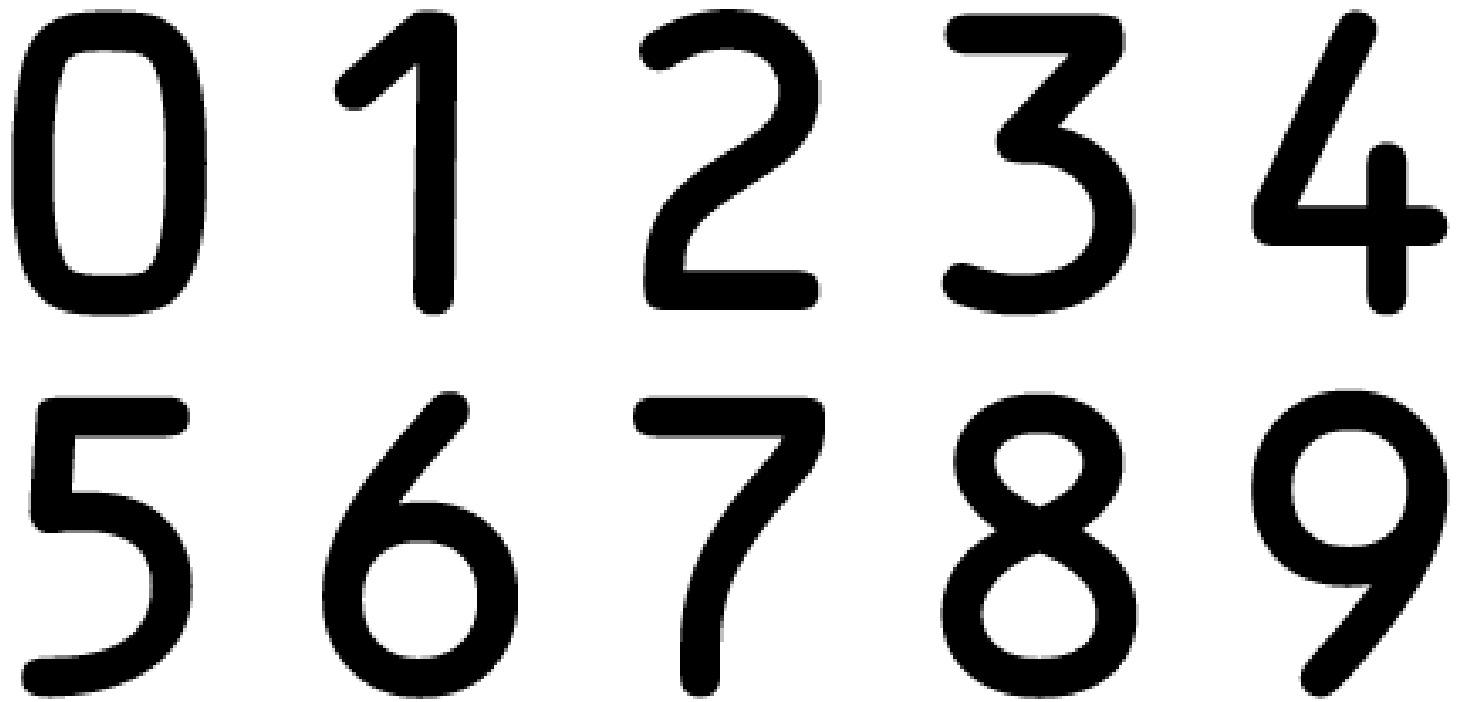
Vegard Antun (UiO)
Anders C. Hansen (Cambridge, UiO)

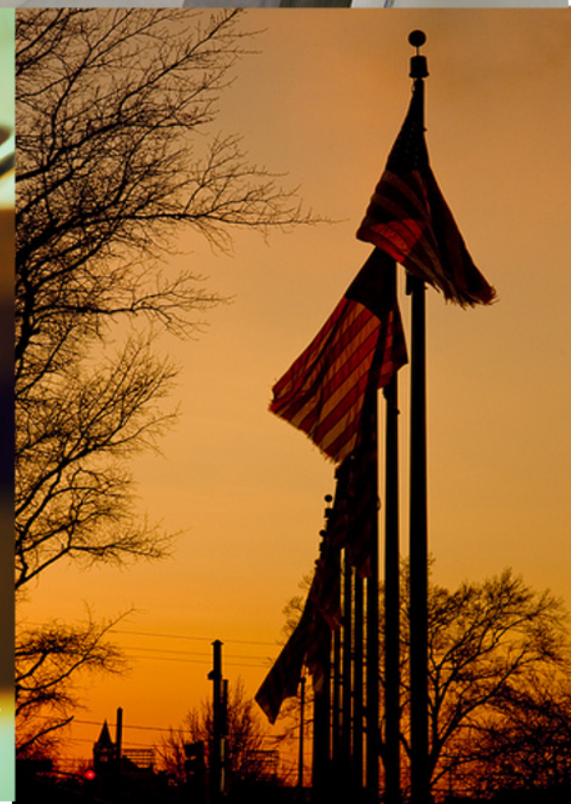
Joint work with:

F. Renna (PU), C. Poon (Cambridge), B. Adcock (SFU)

May 16, 2018

Image classification





ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

Given images $x_1, \ldots, x_n$ assign to each image one of C predefined labels.

Results

1. (Error: 16.4%) Neural Network
2. (Error: 26.1%) Handcrafted feature extraction + classifier
3. (Error: 26.9%) Handcrafted feature extraction + classifier
4. (Error: 29.5%) Handcrafted feature extraction + classifier
5. (Error: 34.4%) Handcrafted feature extraction + classifier

Results

1. (Error: 2.3%) Neural Network
2. (Error: 2.5%) Neural Network
3. (Error: 2.7%) Neural Network
4. (Error: 3.0%) Neural Network
5. (Error: 3.2%) Neural Network

Notation

- $\mathcal{X}$ - Space of images.
- $\mathcal{R}$ - Space of perturbations.
- $T_r : \mathcal{X} \to \mathcal{X}$ - Perturbation operator for $r \in \mathcal{R}$.
- $f : \mathcal{X} \to [0,1]^C$ - Neural network classifier.
- $\hat{k} : \mathcal{X} \to \{1, 2, \ldots, C\}$. $\hat{k} = \text{argmax} \circ f$. Predicted label.

Example 1

Additive noise: Let $\mathcal{R} \subseteq \mathcal{X}$ and $T_r(x) = x + r$.

Notation

Minimal perturbation changing the prediction

$$r^*(x) \in \text{argmin}_{r \in \mathcal{R}} ||r||_{\mathcal{R}} \text{ subject to } \hat{k}(T_r(x)) \neq \hat{k}(x)$$

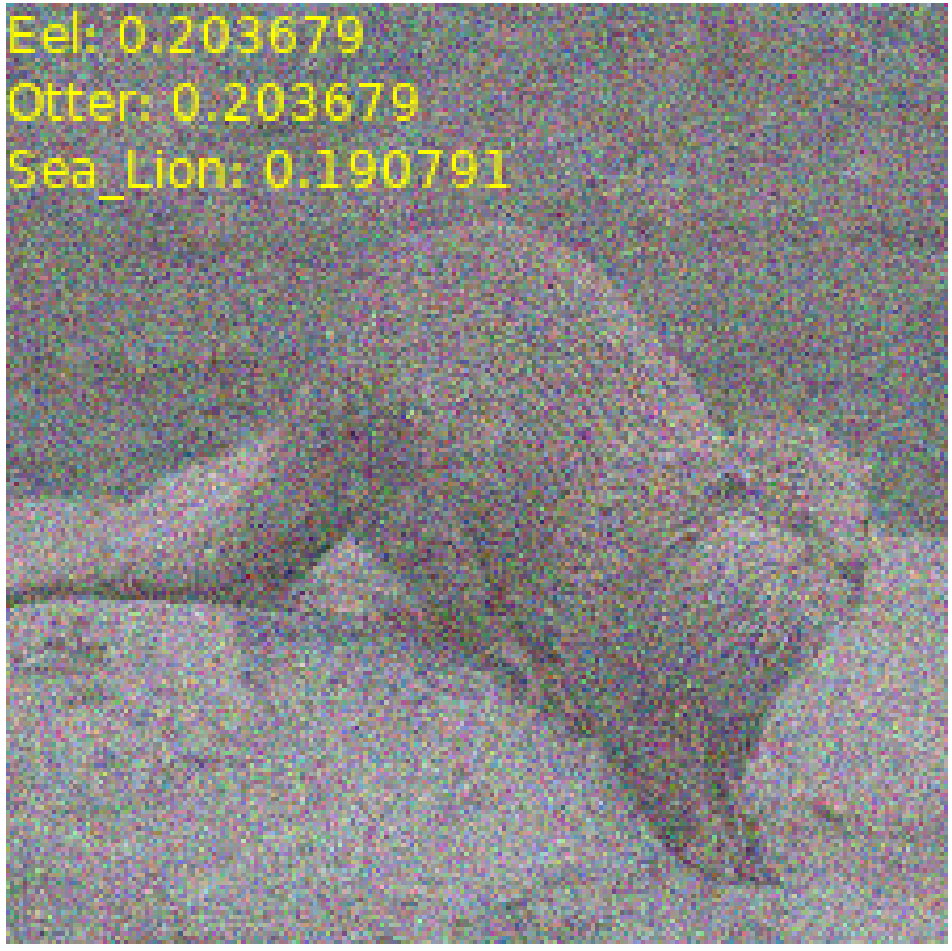

Otter: 0.993142
Beaver: 0.00231697
Mink: 0.00199465

Random Noise

1. Pick $v \in \mathbb{S}^{d-1}$ (d is dimension of $\mathcal{X}$)
2. Solve

$$r^* \in \text{argmin}_{r \in \{\alpha v : \alpha > 0\}} ||r||_2 \quad \text{subject to} \quad \hat{k}(x + r) \neq \hat{k}(x)$$

Random Noise



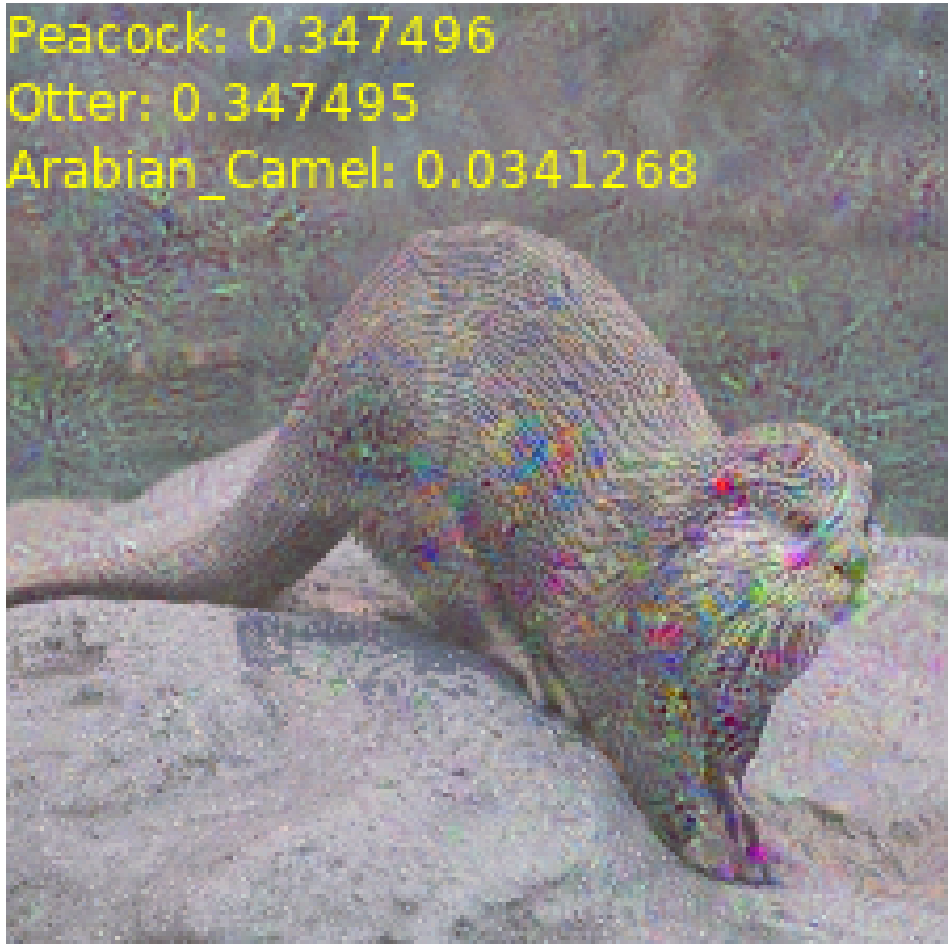
$$||r||_2/||x||_2 = 1.47$$

Fast Gradient

1. Let $L(r) = ||f(x + r) - y_{\text{true}}||_2^2$.
2. Let $v = \nabla_r L$
3. Solve

$$r^* \in \text{argmin}_{r \in \{\alpha v : \alpha > 0\}} ||r||_2 \quad \text{subject to} \quad \hat{k}(x + r) \neq \hat{k}(x)$$

Fast Gradient



$||r||_2/||x||_2 = 0.34$

Deep Fool

$||r||_2/||x||_2 = 0.0052$

Deep Fool for Binary Affine Classifier

Let $f : \mathbb{R}^d \to \mathbb{R}$ be given by $f(x) = w^T x + b$.

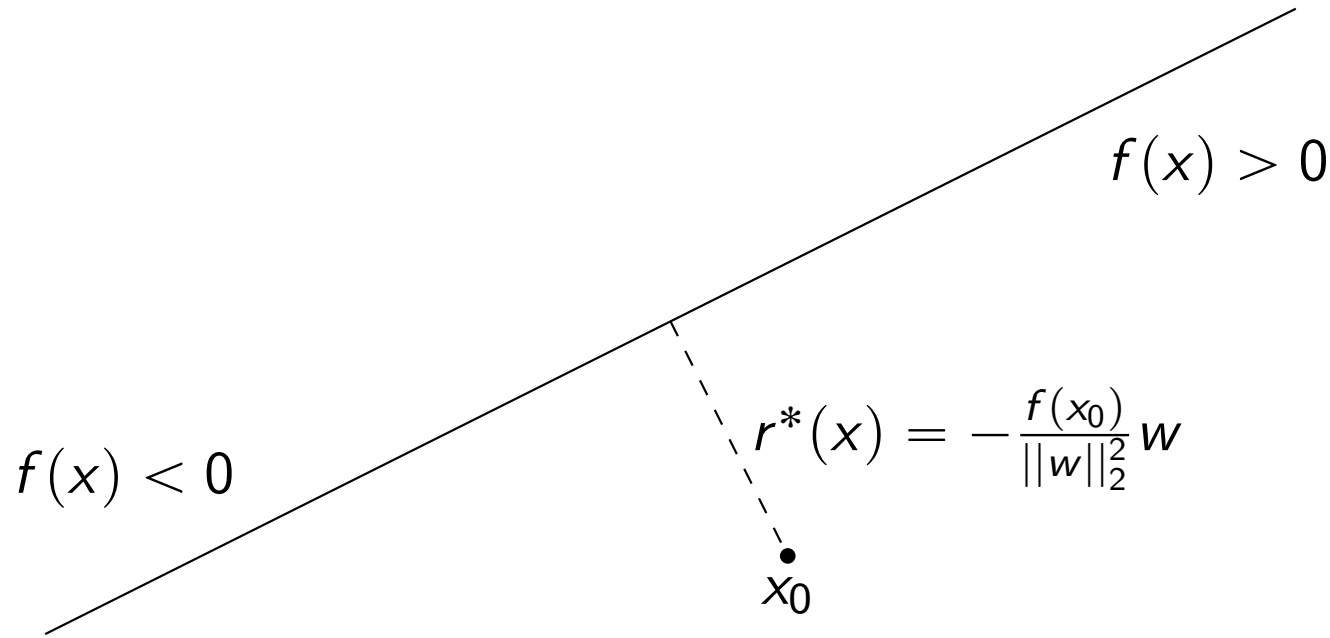


Figure: Adversarial example for a affine binary classifier

Deepfool: a simple and accurate method to fool deep neural networks (2016), S.M. Moosavi-Dezfooli et al.

Deep Fool for Binary Classifier

We would like to solve

$$r_i^*(x_0) \in \text{argmin} \, ||r||_2 \quad \text{subject to} \quad \text{sign}(f(x_0 + r)) \neq \text{sign}(f(x_0))$$

Linerize f by $f(x_i) + \nabla f(x_i)^T (x - x_i)$.

1: **Input:** *image* x, *classifer* f
2: **Output:** *Perturbation* $\tilde{r}$
3: Initialize: $x_0 \leftarrow x, i \leftarrow 0, \eta \leftarrow 0.02, \tilde{r} \leftarrow 0$
4: **while** $\text{sign}(f(x + (1 + \eta)\tilde{r})) = \text{sign}(f(x))$ **do**
5: $\quad r_i \leftarrow -\frac{f(x_i)}{||\nabla f(x_i)||_2} \nabla f(x_i)$
6: $\quad x_{i+1} \leftarrow x_i + r_i$
7: $\quad \tilde{r} \leftarrow \tilde{r} + r_i$
8: $\quad i \leftarrow i + 1$
9: **return** $\tilde{r}$

Deep Fool for Multiclass Affine Classifier

Let $f : \mathbb{R}^d \to \mathbb{R}^C$ be given by

$$f(x) = W^T x + b$$

i.e. $W \in \mathbb{R}^{d \times C}$ and $b \in \mathbb{R}^C$.

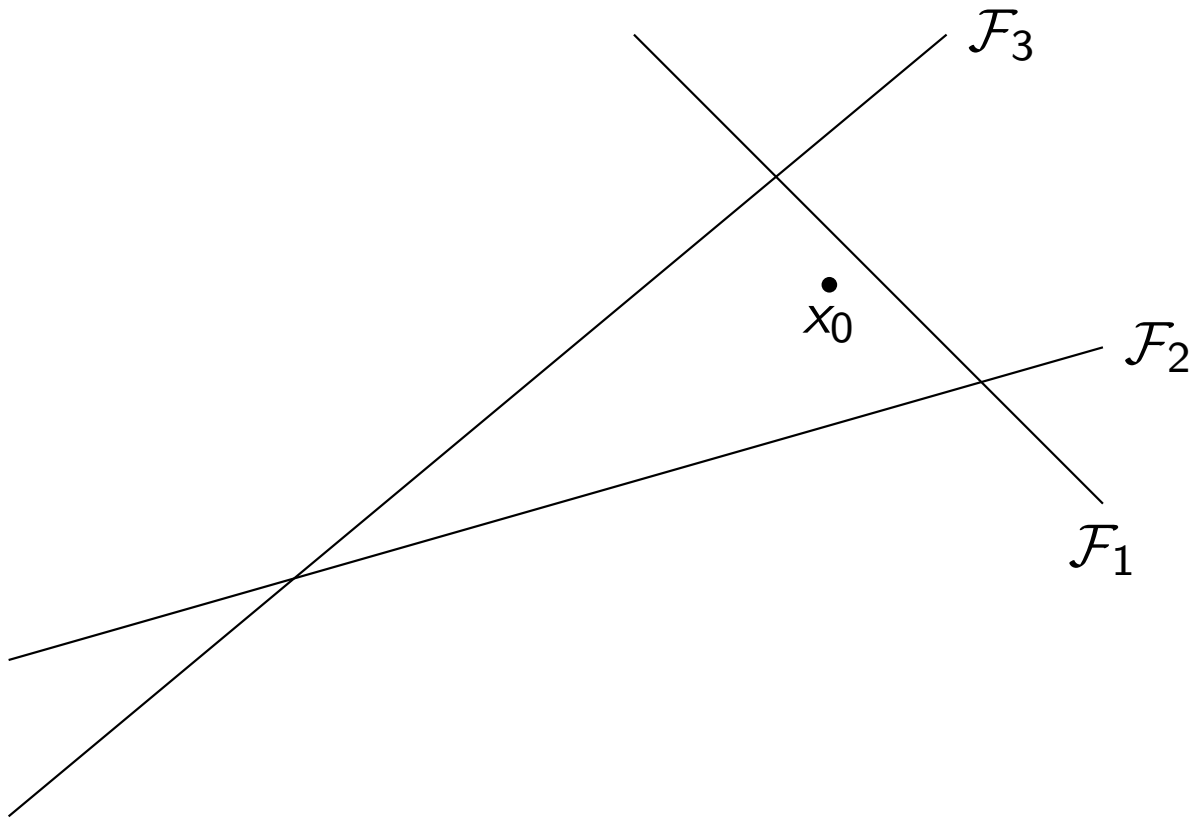
Solve

$$\text{argmin}_r \, ||r||_2 \quad \text{s.t.} \quad \exists k : w_k^T (x_0 + r) + b_k \geqslant w_{\hat{k}(x_0)}^T (x_0 + r) + b_{\hat{k}(x_0)}$$

Deep Fool for Multiclass Affine Classifier

Let

$$P = \bigcap_{k=1}^{C} \{x : f_{\hat{k}(x_0)}(x) \geqslant f_k(x)\}$$



$$\mathcal{F}_k = \{x : f_{\hat{k}(x_0)}(x) - f_k(x) = 0\}$$

Deep Fool for Multiclass Affine Classifier

The problem

$$\text{argmin}_r \, ||r||_2 \quad \text{s.t.} \quad \exists k : w_k^T(x_0 + r) + b_k \geqslant w_{\hat{k}(x_0)}^T(x_0 + r) + b_{\hat{k}(x_0)}$$

has the solution

$$\hat{l}(x_0) = \text{argmin}_{k \neq \hat{k}(x_0)} \frac{|f_k(x_0) - f_{\hat{k}(x_0)}(x_0)|}{||w_k - w_{\hat{k}(x_0)}||_2}$$

$$r^*(x_0) = \frac{|f_{\hat{l}(x_0)}(x_0) - f_{\hat{k}(x_0)}(x_0)|}{||w_{\hat{l}(x_0)} - w_{\hat{k}(x_0)}(x_0)||_2^2} \left(w_{\hat{l}(x_0)} - w_{\hat{k}(x_0)} \right)$$

Deep Fool for Multiclass Classifier

Idea: Linerize f as $f(x_i) + \nabla f(x_i)^T (x - x_i)$

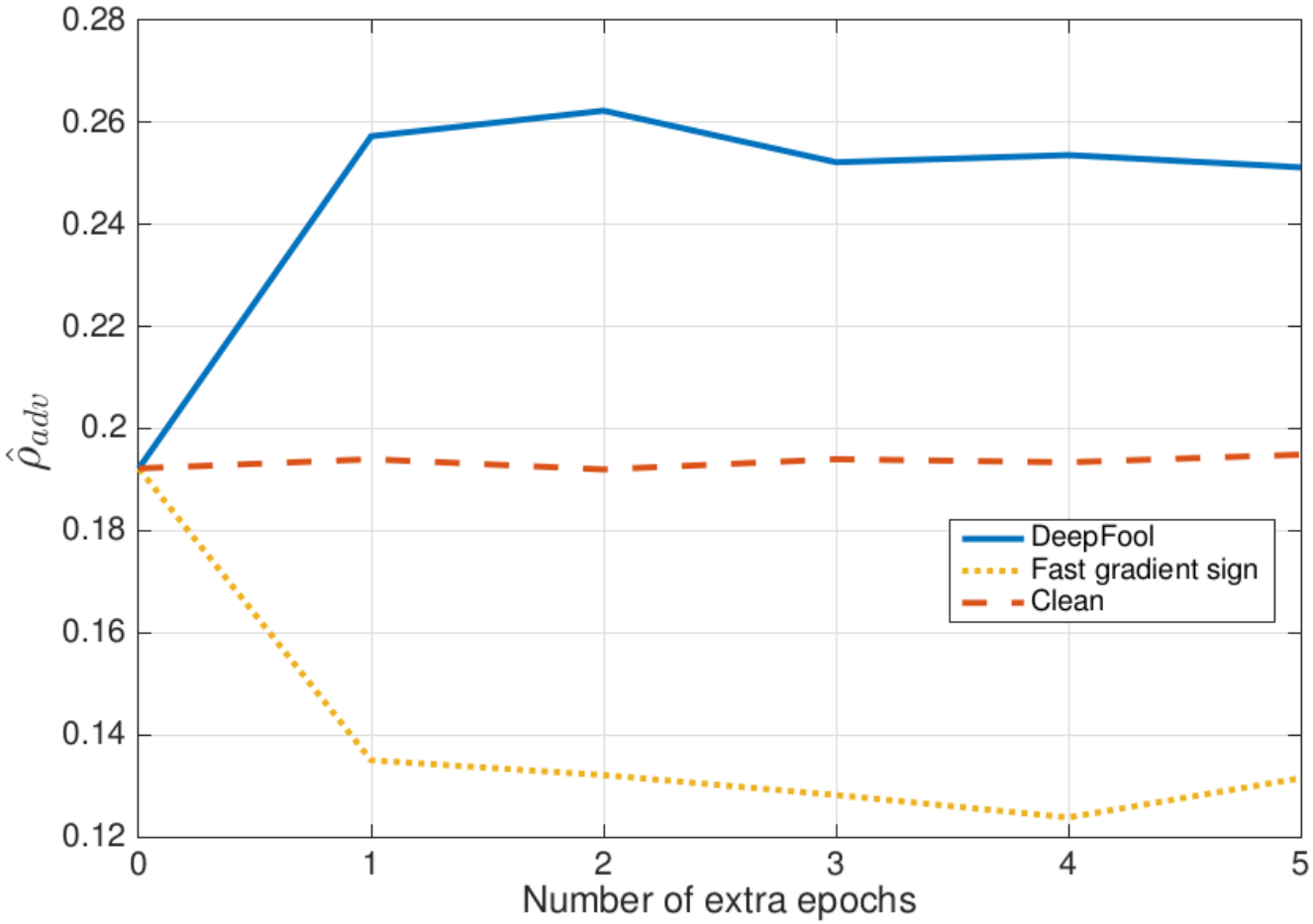
At each iteration solve the problem

$\text{argmin}_r ||r||_2 \quad \text{s.t.} \quad \exists k : \nabla f_k(x_i)^T r + f_k(x_i) \geqslant \nabla f_{\hat{k}(x_0)}(x_i)^T r + f_{\hat{k}(x_0)}(x_i)$

and make the update

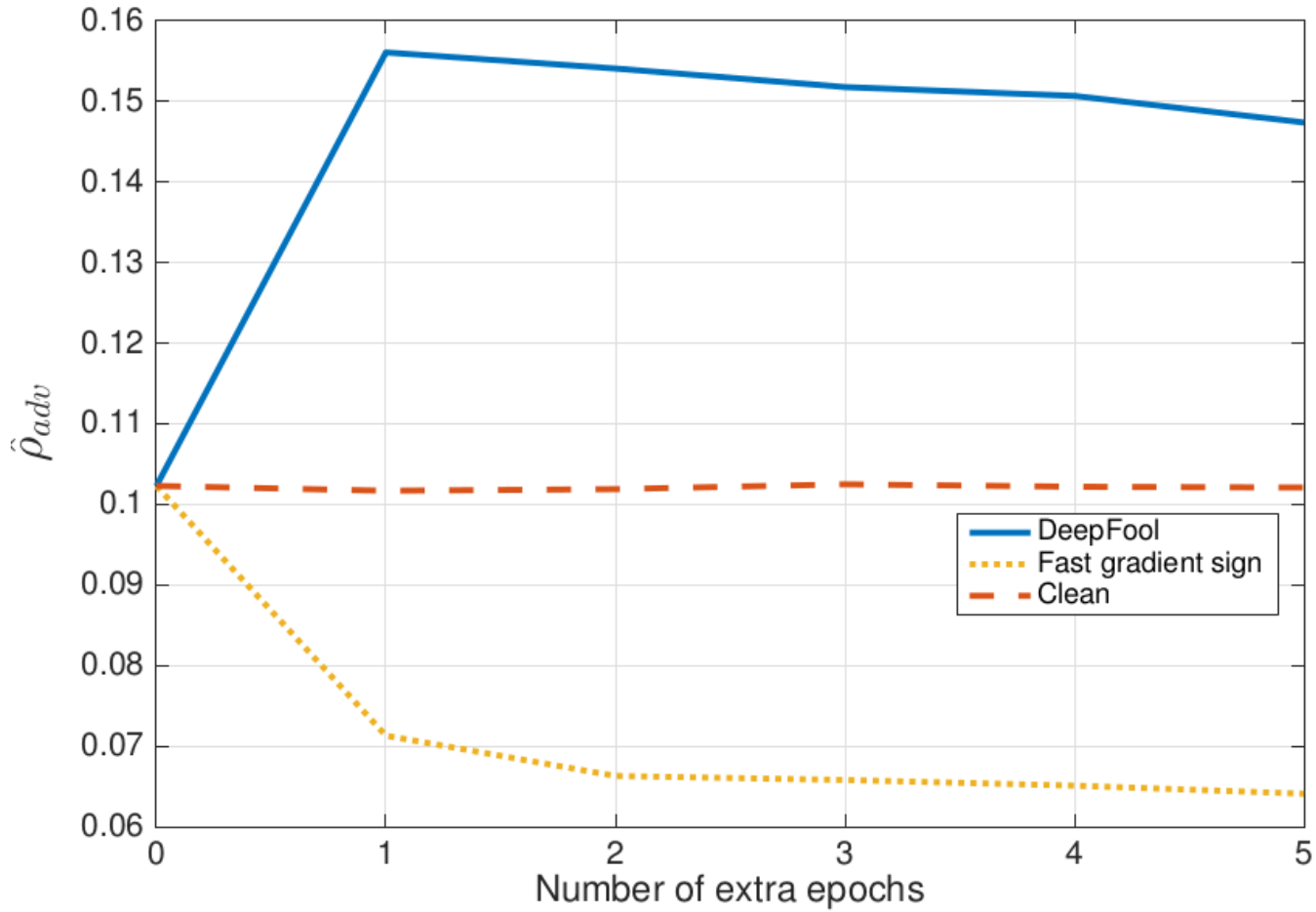
$$x_{i+1} \leftarrow x_i + r_i$$

Training on adversarial examples



$$\rho_{\text{adv}} = \frac{1}{n} \sum_{i=1}^{n} \frac{||r^*(x_i)||}{||x_i||}$$

Training on adversarial examples



$$\rho_{\text{adv}} = \frac{1}{n} \sum_{i=1}^{n} \frac{||r^*(x_i)||}{||x_i||}$$

Universal perturbations

Definition 2 (General)

Let $\xi \geqslant 0$, $x \in \mathcal{X}$ and $r \in \mathcal{R}$. We call r an universial perturbation if

(i) $||r||_{\mathcal{R}} \leqslant \xi$,

(ii) $\hat{k}(T_r(x)) \neq \hat{k}(x)$ "for most" x.

Universal perturbations

Definition 3

Let $0 < \delta < 1$ and $\{x_1, \ldots, x_m\} = X \subset \mathcal{X}$. We will search for a perturbation $r \in \mathcal{R} = \mathcal{X}$ such that

(i) $||r||_2 \leqslant \xi$

(ii)

$$\text{Err}(X, r) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{\hat{k}(x_i + r) \neq \hat{k}(x_i)} \geqslant 1 - \delta$$

Universal perturbations

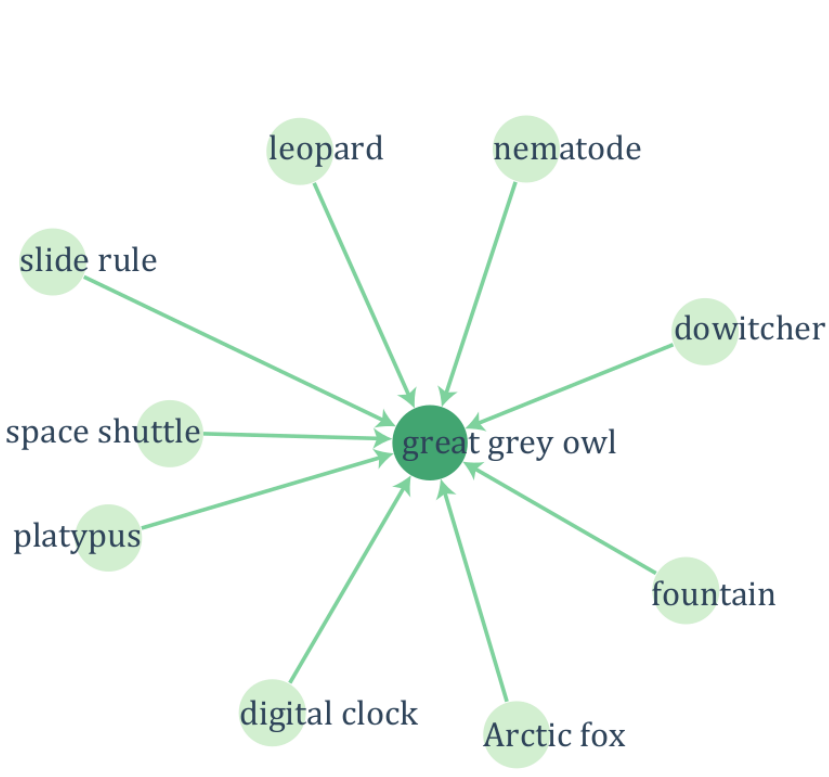
Input: *images* $X = \{x_1, \ldots, x_m\}$, *classifer* f, *fooling ratio* δ, *norm of perturbation* ξ
Output: *Univeral perturbation* r
Initialize: $r \leftarrow 0$
while $\text{Err}(X, r) < 1 - \delta$ **do**
    for each $x_i \in X$ **do**
        $\Delta r_i \leftarrow \text{argmin}_v \, ||v||_2$ s.t. $\hat{k}(x_i + r + v) \neq \hat{k}(x_i)$
        $r \leftarrow (r + \Delta r_i) \min\left(1, \frac{\xi}{||r + \Delta r_i||_2}\right)$
return r

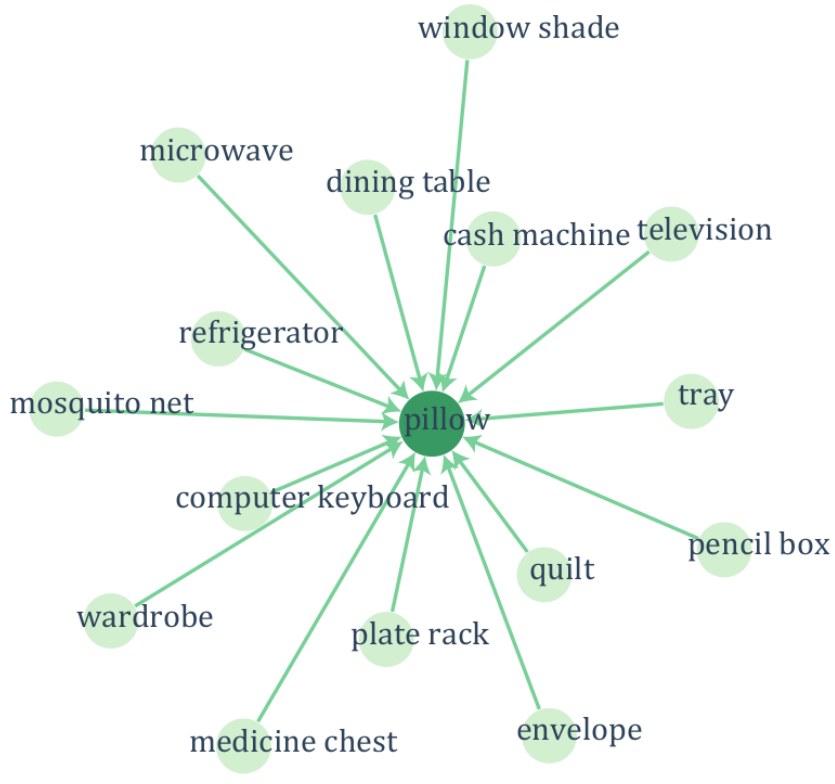
Universal Perturbation for large networks

	VGG-F	CaffeNet	GoogLeNet	VGG-16	VGG-19	ResNet-152
VGG-F	**93.7%**	71.8%	48.4%	42.1%	42.1%	47.4%
CaffeNet	74.0%	**93.3%**	47.7%	39.9%	39.9%	48.0%
GoogLeNet	46.2%	43.8%	**78.9%**	39.2%	39.8%	45.5%
VGG-16	63.4%	55.8%	56.5%	**78.3%**	73.1%	63.4%
VGG-19	64.0%	57.2%	53.6%	73.5%	**77.8%**	58.0%
ResNet-152	46.3%	46.3%	50.5%	47.0%	45.5%	**84.0%**

Table: The rows indicate the architecture for which the universal perturbations is computed, and the columns indicate the architecture for which the fooling rate is reported.

Universal adversarial perturbations (2017) S.M. Moosavi-Dezfooli et al.

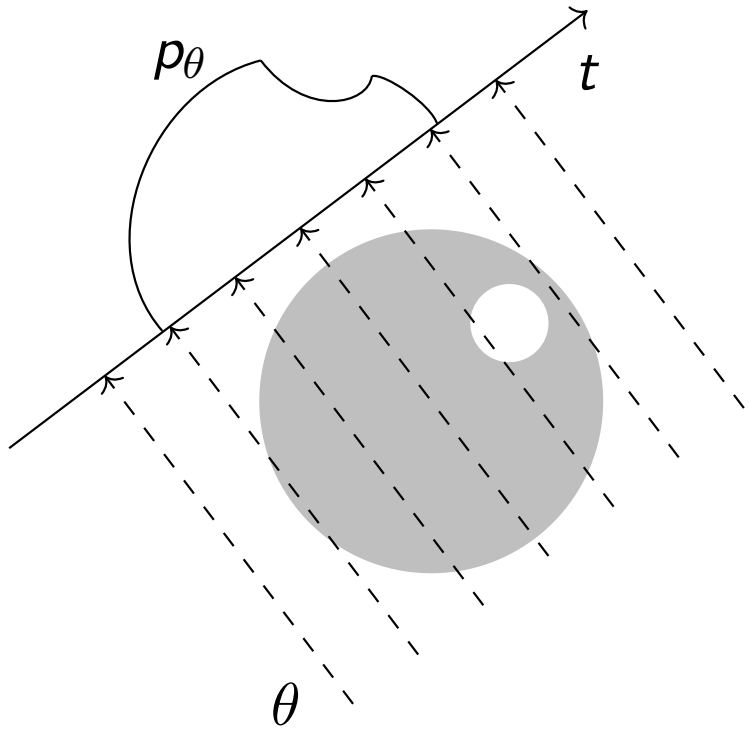

leopard
nematode
slide rule
dowitcher
space shuttle
great grey owl
platypus
fountain
digital clock
Arctic fox


window shade
microwave
dining table
cash machine
television
refrigerator
mosquito net
pillow
tray
computer keyboard
pencil box
quilt
wardrobe
plate rack
medicine chest
envelope

Radon Sampling

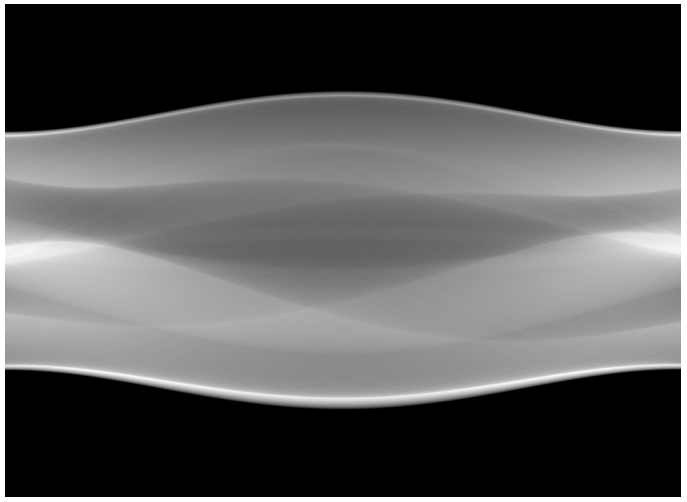
Let $\theta \in [0, \pi)$ and $t \in \mathbb{R}$ the radon transform of $f \in L^2(\mathbb{R}^2)$ is

$$p_\theta(t) = \int\int f(x, y)\delta(x\cos\theta + y\sin\theta - t)dxdy$$

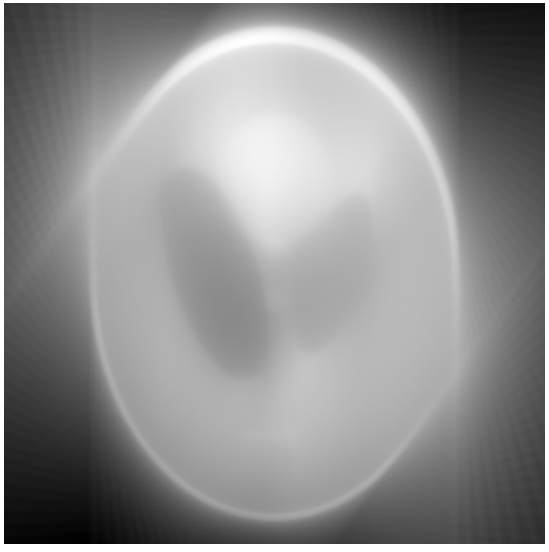


Filtered Backprojection (FBP)

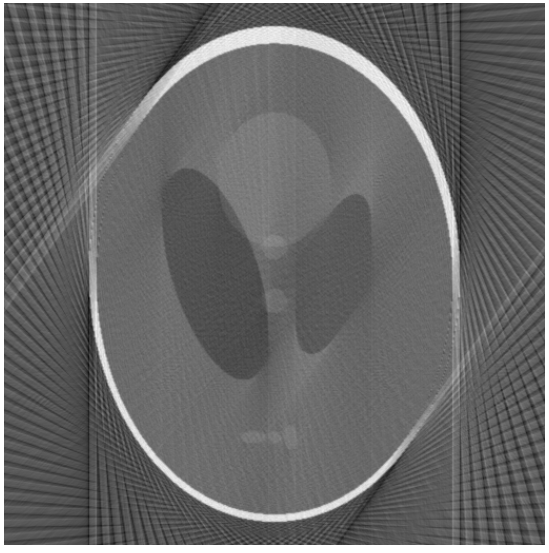
Image x



Radon transform $y = Ax$



Adjoint A^*y



Filtered backproj. $\tilde{A}^{-1}y$

Fourier Sampling

Let $N = 2^r$ and suppose $f \in \{e^{2\pi i(\omega_1 t_1 + \omega_2 t_2)}/N\}_{\omega_1,\omega_2=-N/2+1}^{N/2}$ i.e

$$f(t_1, t_2) = \sum_{\omega_1=-N/2+1}^{N/2} \sum_{\omega_2=-N/2+1}^{N/2} y_{\omega_1,\omega_2} \frac{1}{N} e^{2\pi i(\omega_1 t_1 + \omega_2 t_2)}$$

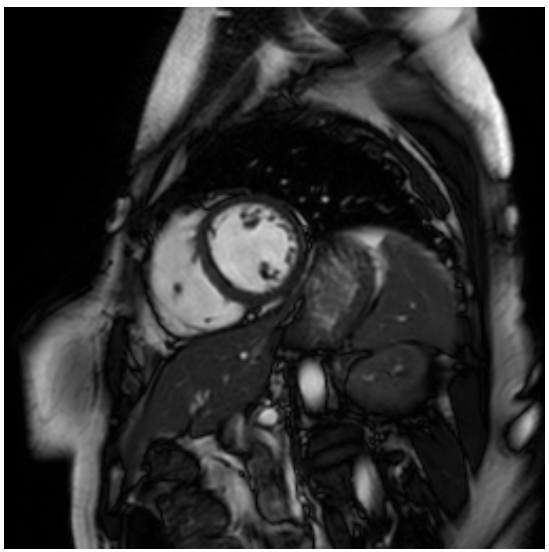
Let $x_{k,l} = f(k/N, l/N)$ for $k, l \in \{0, \ldots, N-1\}$. Then

$$y = Fx$$

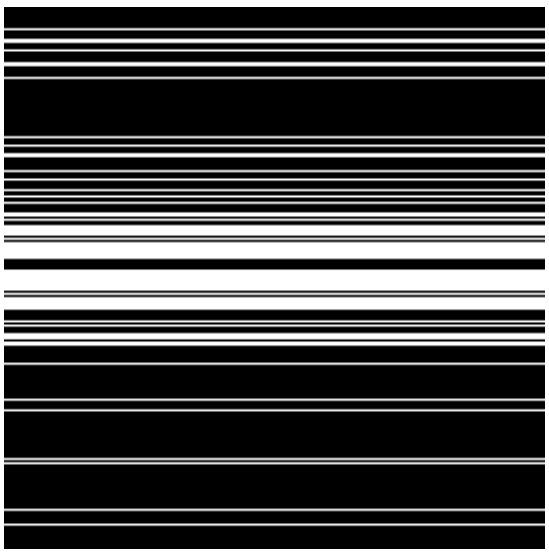
where $F \in \mathbb{C}^{N^2 \times N^2}$ is the Fourier matrix. This matrix is unitary.

Fourier Sampling

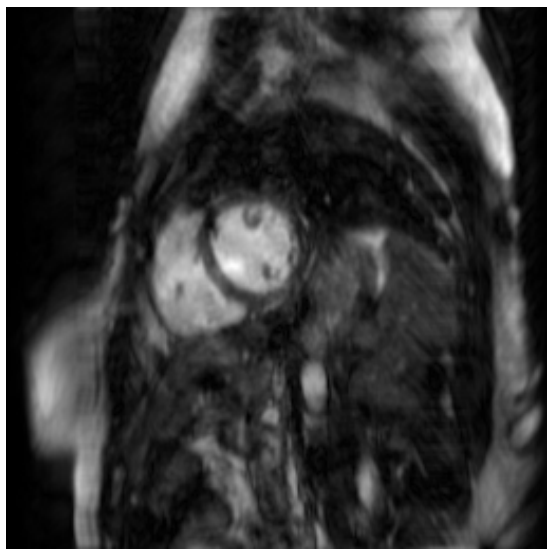
Let $A = P_\Omega F$ and $y = Ax$.



Original x



Sampling pattern Ω



Adjoint: A^*y

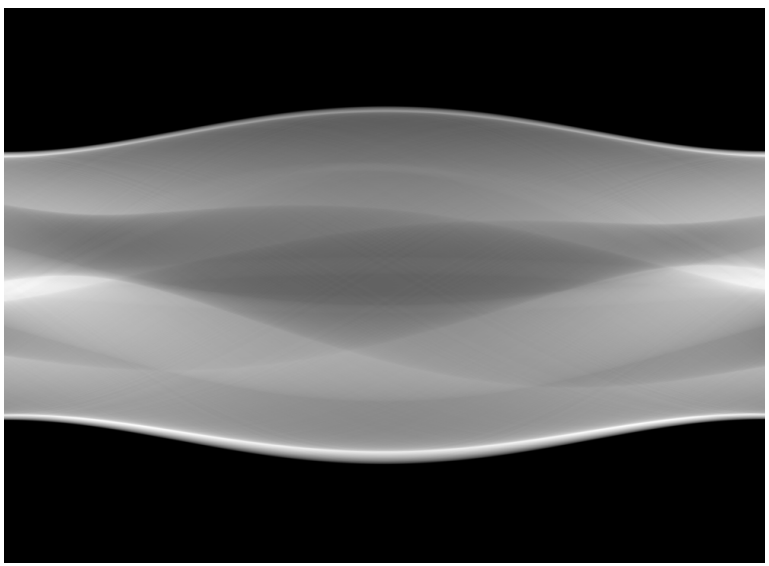
Fourier Slice Theorem

Theorem 4

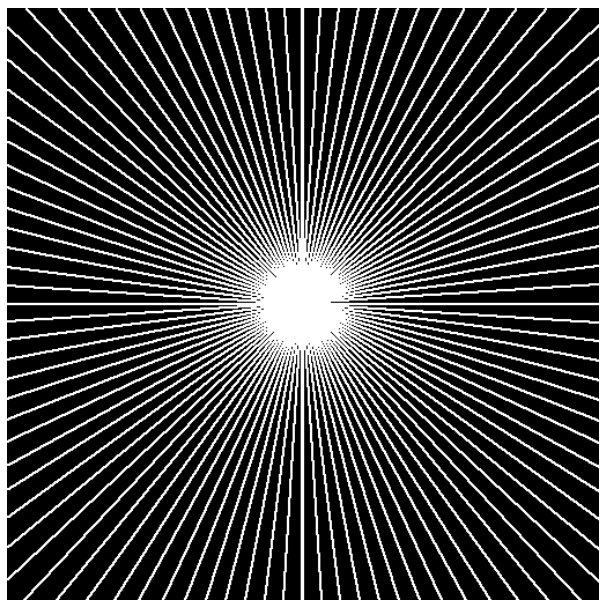
Let $f \in L^2(\mathbb{R}^2)$. Then for all $\theta \in [0, \pi)$ and all $\xi \in \mathbb{R}$ the Fourier transform of f satisfies

$$\hat{p}_\theta(\xi) = \hat{f}(\xi \cos\theta, \xi \sin\theta)$$

where $\hat{p}_\theta$ and $\hat{f}$ are the Fourier transform of p_θ and f, respectively.



(a) Radon sampling



(b) Fourier sampling pattern $\Omega \subset \{1, \ldots, N^2\}$

Compressive Sensing Summary

Let $A \in \mathbb{C}^{m \times N}$ with $m < N$ and let $x \in \mathbb{C}^N$ be some unknown signal. Let $W \in \mathbb{R}^{N \times N}$ be a sparsifying transform. Let $y = AW^{-1}x + e$ where $e \in \mathbb{C}^m$ is some measurement error satisfying $||e||_2 \leqslant \eta$. If AW^{-1} satisfies the RIPL then any solution $\hat{x}$ to the optimization problem

$$\text{minimize}_{z \in \mathbb{C}^N} ||z||_1 \text{ subject to } ||AW^{-1}z - y||_2 \leqslant \eta$$

satisfies

$$||\hat{x} - x||_2 \lesssim \frac{\sigma_{\mathbf{s},\mathbf{M}}(x)_1}{\sqrt{s}} + \eta$$

where

$$\sigma_{\mathbf{s},\mathbf{M}}(x)_1 = \inf\{||x - z||_1 : z \text{ is } (\mathbf{s}, \mathbf{M})\text{-sparse}\}$$

Why use Neural Networks for Inverse Problems?

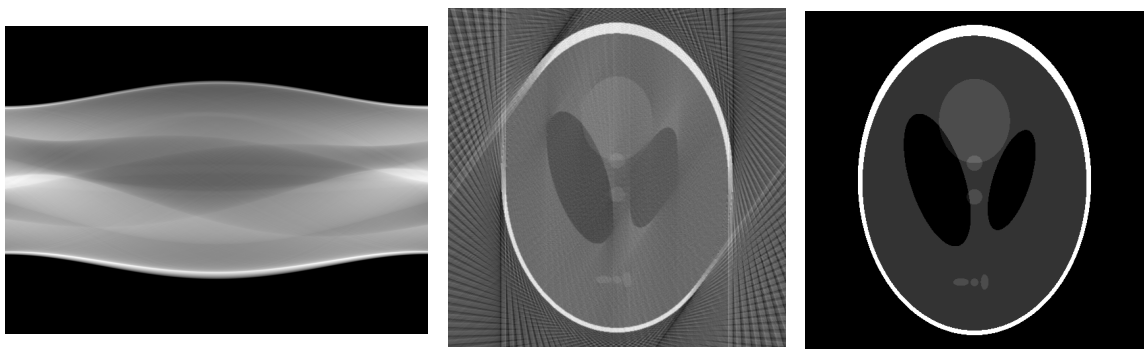
- Neural networks are fast!
- Neural networks preformes better?

Neural Networks for Inverse Problems

- Recall $y = Ax$, $y \in \mathbb{C}^m$ and $x \in \mathbb{C}^N$.
- In general difficult to create a Neural Network map directly from $\mathbb{C}^m \to \mathbb{C}^N$.
- Convolutional Neural Network are often on the form

$$f(y) = A^* y - G(A^* y)$$

where
$G(x) = \rho(W_L \rho(W_{L-1}(\cdots \rho(W_1 z + b_1) \cdots) + b_{L-1}) + b_L)$



y $\tilde{A}^{-1} y$ x

Neural Networks for Inverse Problems

- Promising idea: Build a neural network using a convex optimization solver.
- Let $A \in \mathbb{C}^{m \times N}$ be the sampling matrix and $W \in \mathbb{C}^{N \times N}$ a convolutional operator.
- Pick you favourite convex optimization algorithm and try to solve

$$\text{minimize} \, ||Wz||_1 \quad \text{subject to} \quad ||Az - y||_2 \leqslant \eta$$

using L iterations.

Different Types of Stabilities

1. Stability w.r.t. adversarial noise.
2. Stability w.r.t. more samples.
3. Stability w.r.t. structual changes.
4. Stability w.r.t. new types of images.

Can we use Deep Fool?

Recall Deep Fool was computing $\nabla f(x_i)$

What does adversarial noise mean for inverse problmes, when $f : \mathbb{C}^m \to \mathbb{C}^N$?

Adversarial noise Inverse Problems

Let $f : \mathbb{C}^m \to \mathbb{C}^N$ be a neural network and let

$$Q(r) = ||f(y + Ar) - f(y)||_2^2 - \lambda||r||_2^2$$

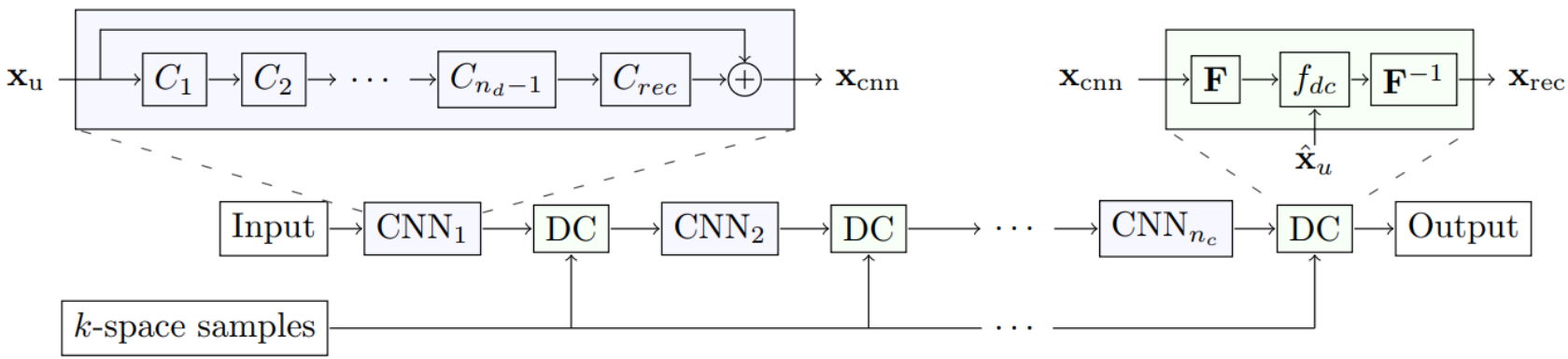
where $r \in \mathbb{C}^N$ and $\lambda > 0$.

Input: *image* x, *classifer* f, *sampling matrix* A
Output: *Perturbation* $r_{\text{max\_iterations}}$
Initialize: $v = 0, i = 0,\ r_0 \in \mathbb{S}^{N-1},\ 0 < \lambda, \gamma, \eta$
while $i \leqslant$ max_iterations **do**
    $v_{i+1} \leftarrow \gamma v_i + \eta \nabla_r Q(r_i)$
    $r_{i+1} \leftarrow r_i + v_{i+1}$
    $i \leftarrow i + 1$
return $r_{\text{max\_iterations}}$

Deep MRI Net

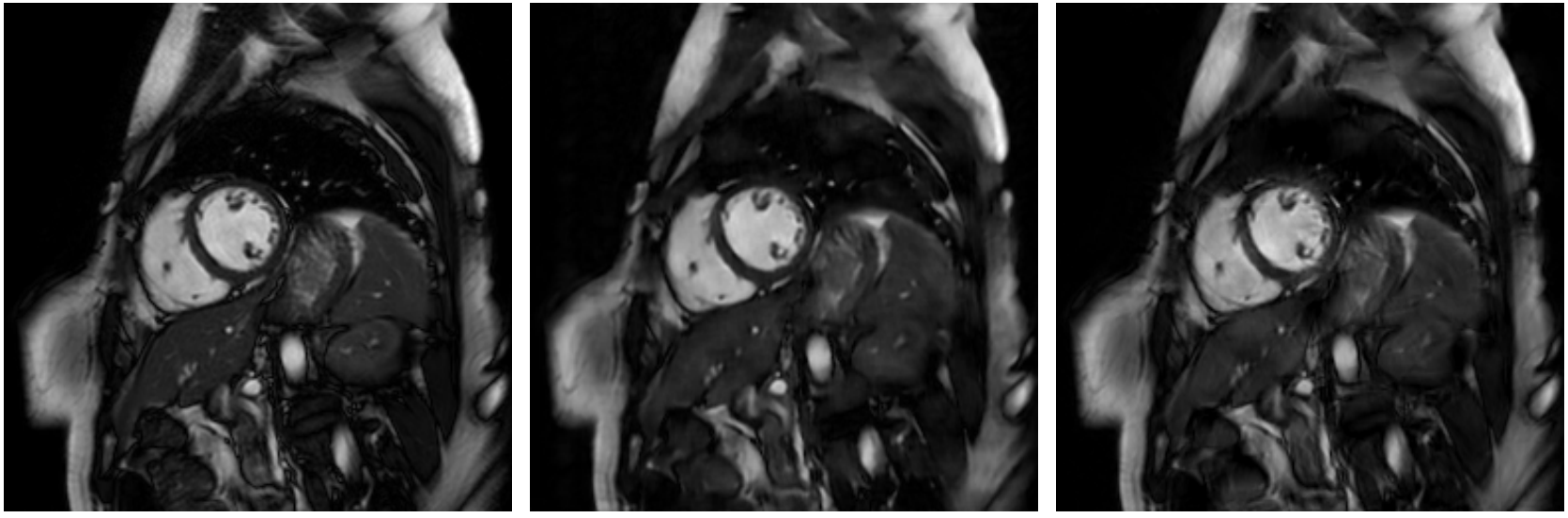
- **Input:** $x_0 \leftarrow A^* y$, where $A = P_\Omega F$.
- $CNN_i(x) = x - G_i(x)$ where G_i is a convolutional neural network.
- $DC(x) = F^{-1}\phi(Fx)$, where

$$\phi(z) = \begin{cases} z_k & \text{if } k \notin \Omega \\ \frac{z_k + \lambda y_k}{1+\lambda} & \text{if } k \in \Omega \end{cases}.$$



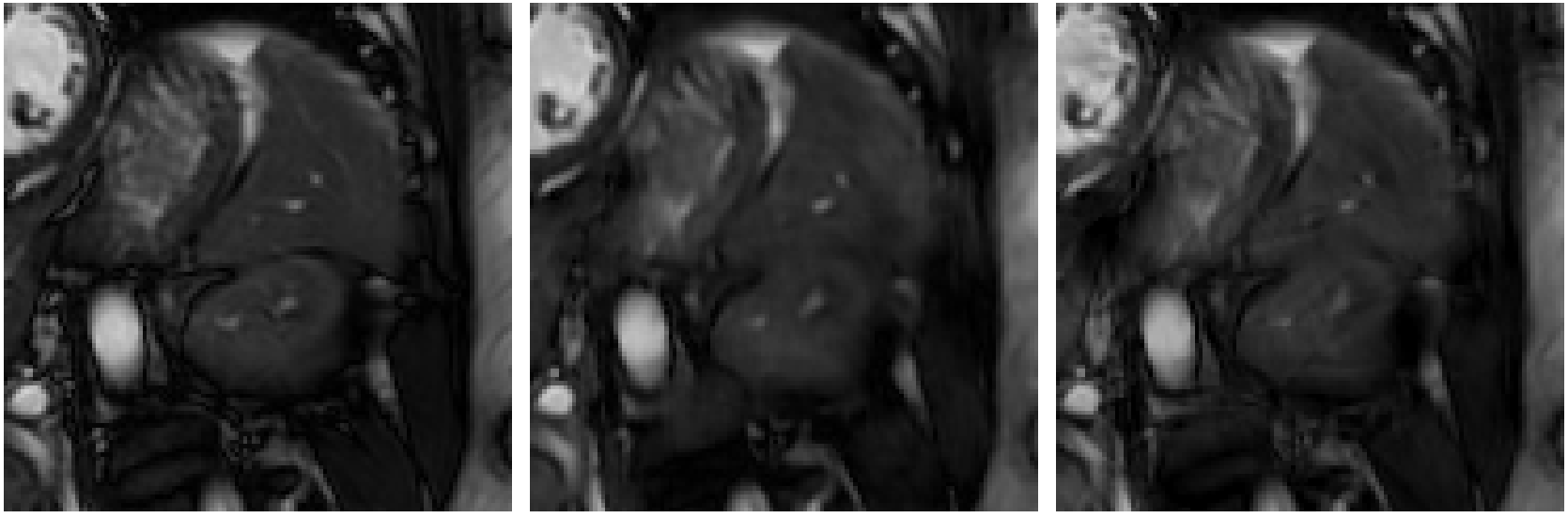
A Deep Cascade of Convolutional Neural Networks for Dynamic MR Image Reconstruction", J. Schlemper, J. Caballero, J. Hajnal, A. Price, D. Rueckert IEEE Trans. Med. Imag. (to appear)

Deep MRI Net



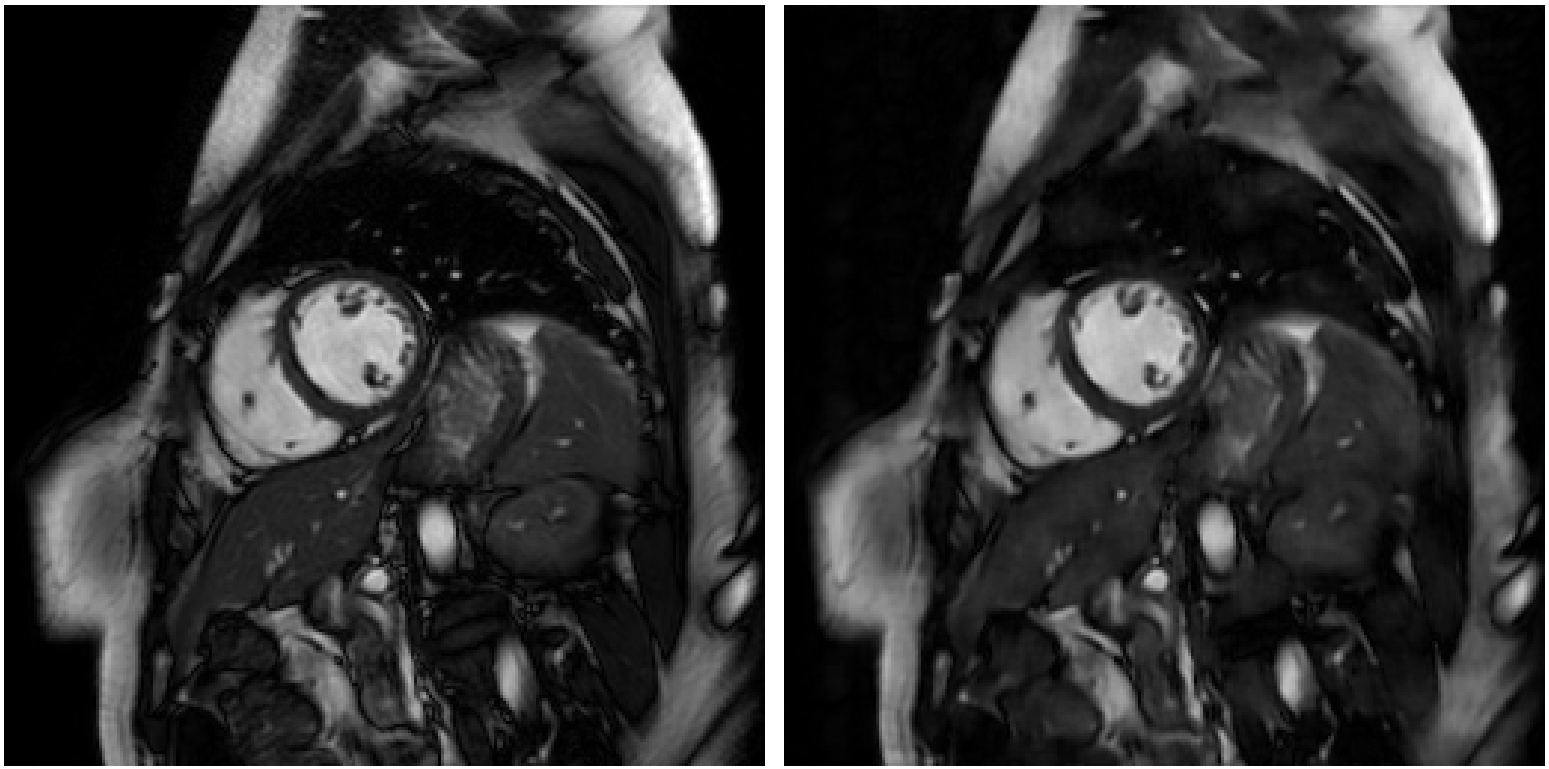
Original | Deep MRI Net | Compressive Sensing

Deep MRI Net



Original | Deep MRI Net | Compressive Sensing

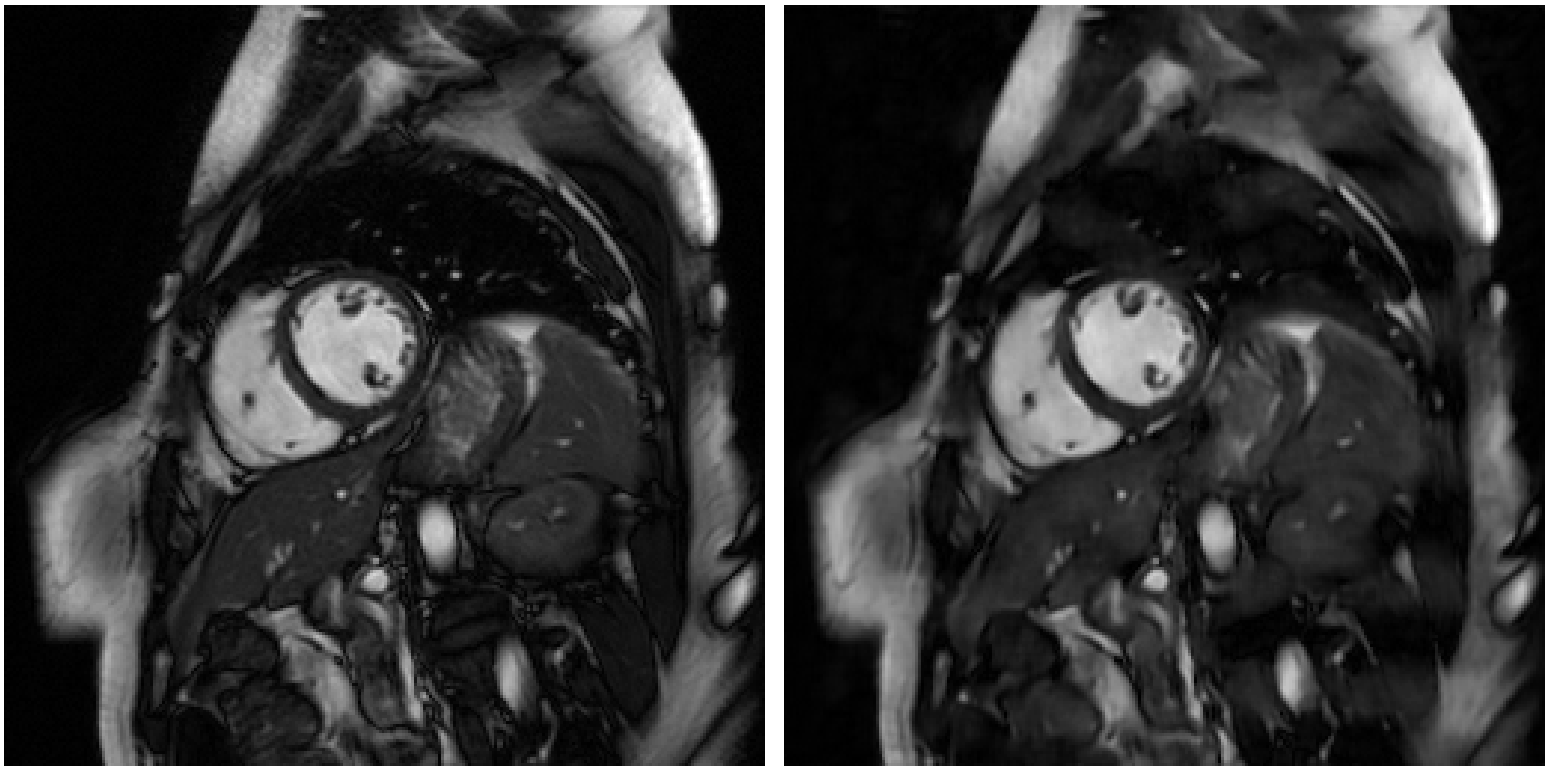
Deep MRI Net – Adversarial Noise



(a) x (b) $f(Ax)$

Figure: Iterations: 0

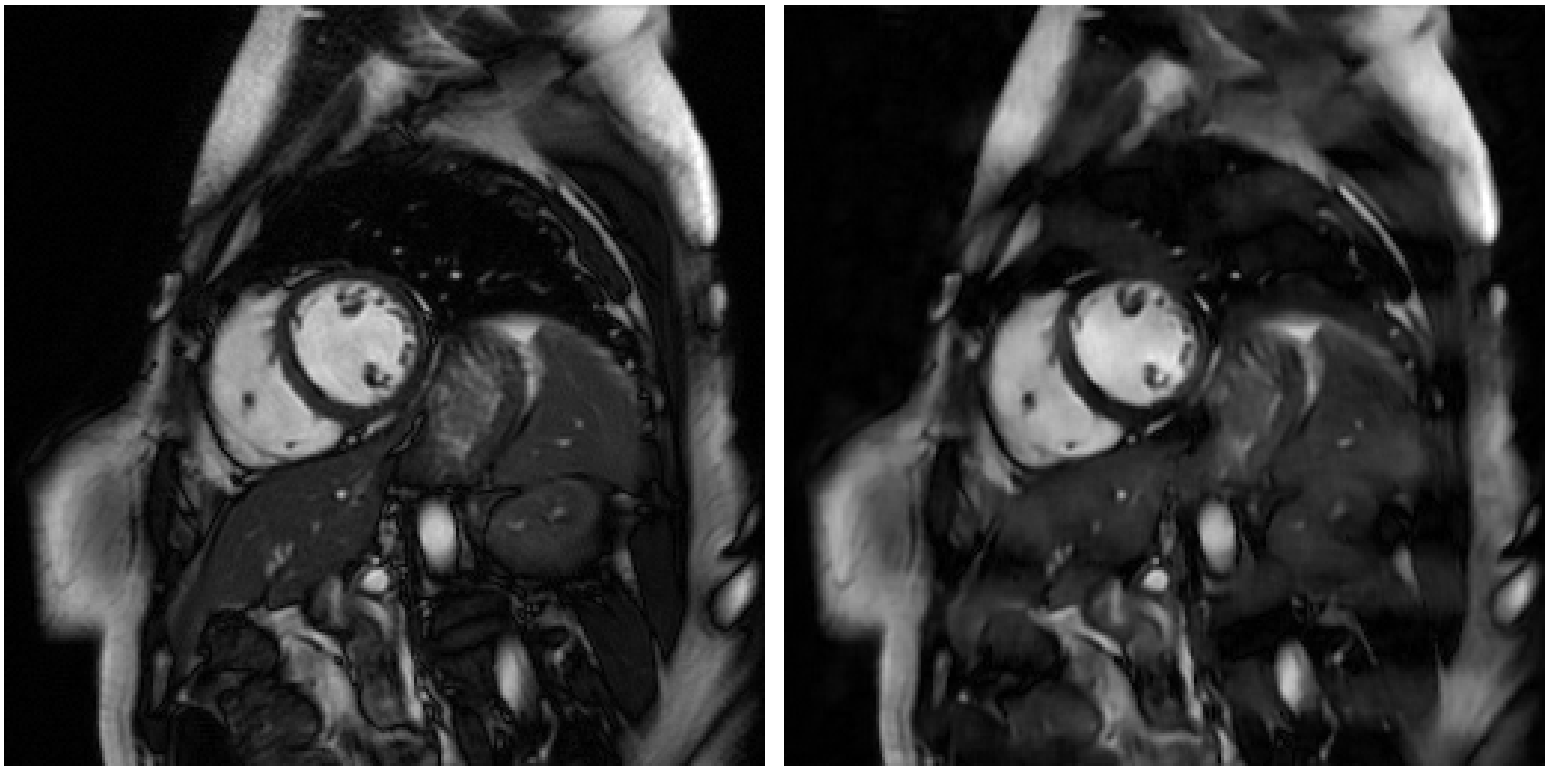
Deep MRI Net – Adversarial Noise



(a) $x + r$ (b) $f(A(x + r))$

Figure: Iterations: 500

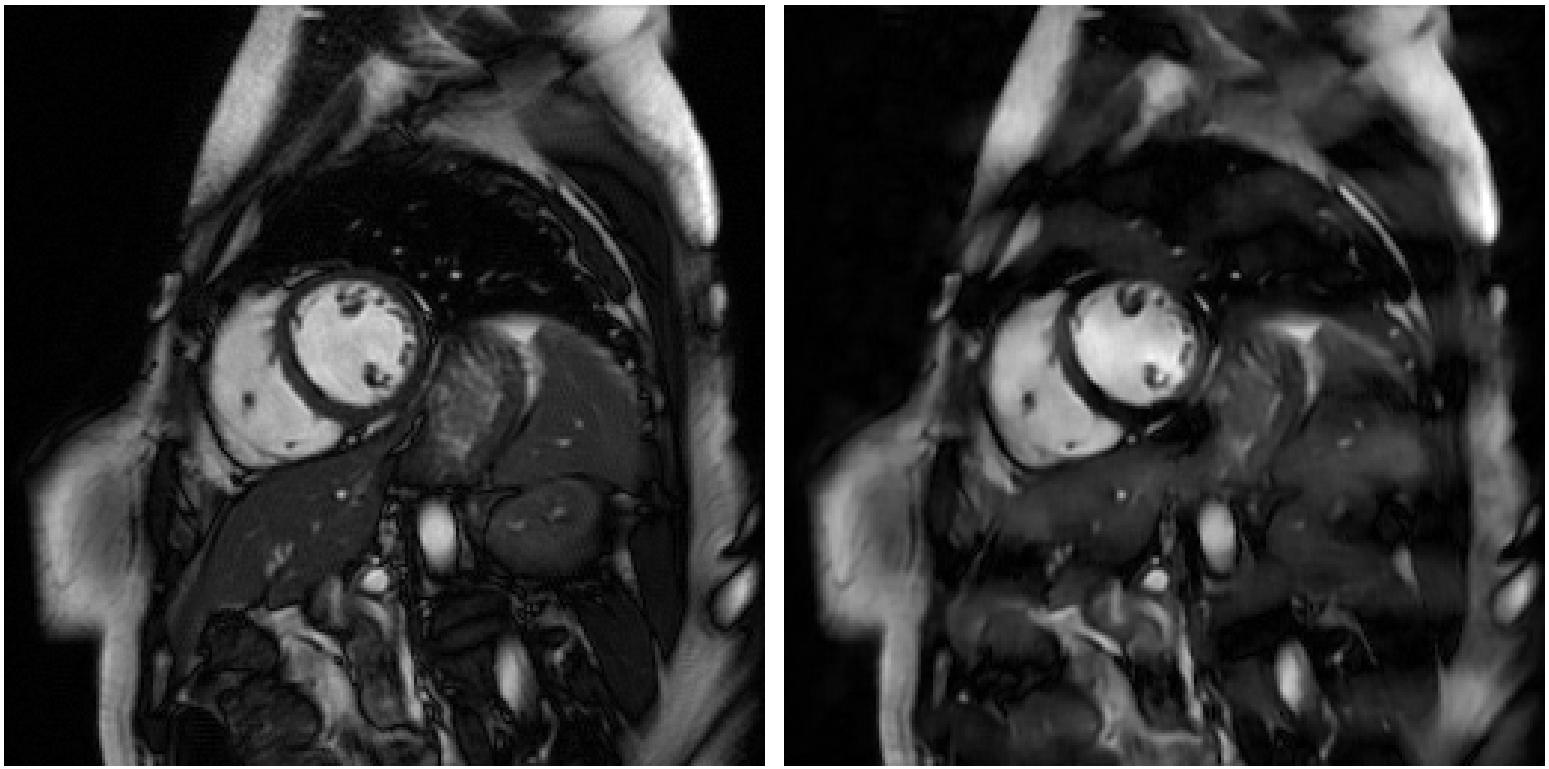
Deep MRI Net – Adversarial Noise



(a) $x + r$ (b) $f(A(x + r))$

Figure: Iterations: 1000

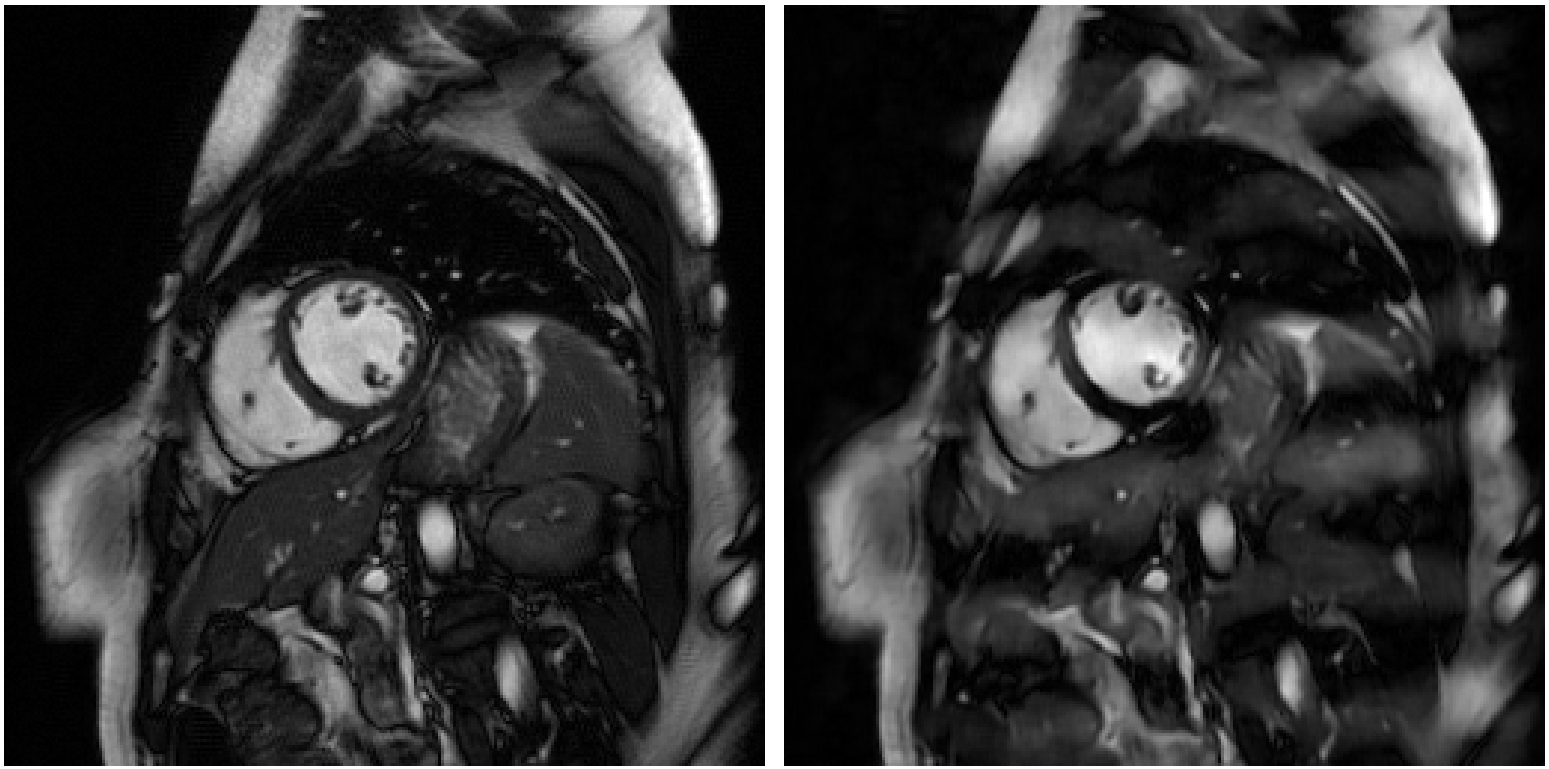
Deep MRI Net – Adversarial Noise



(a) $x + r$ (b) $f(A(x + r))$

Figure: Iterations: 1500

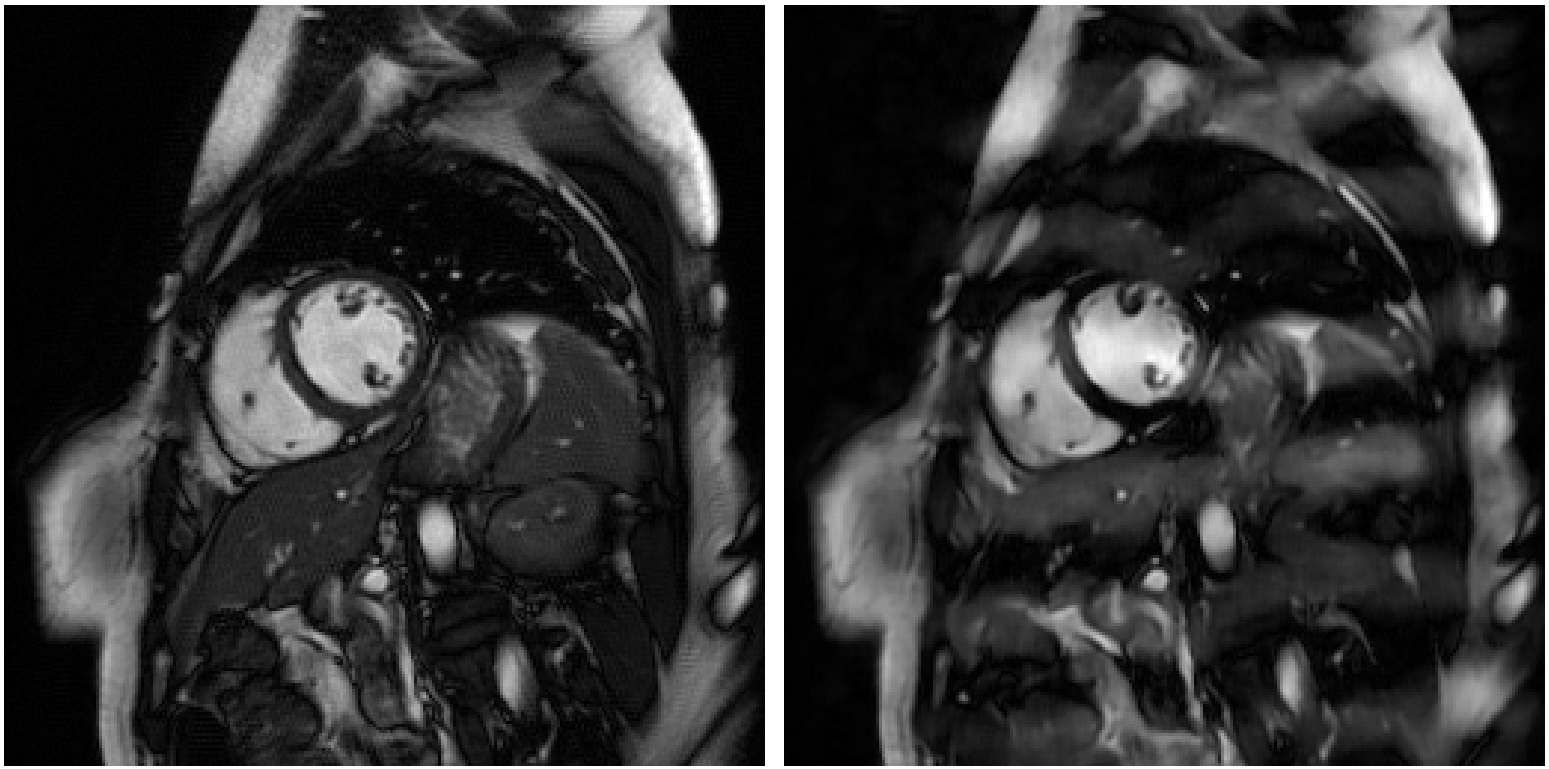
Deep MRI Net – Adversarial Noise



(a) $x + r$ (b) $f(A(x + r))$

Figure: Iterations: 2000

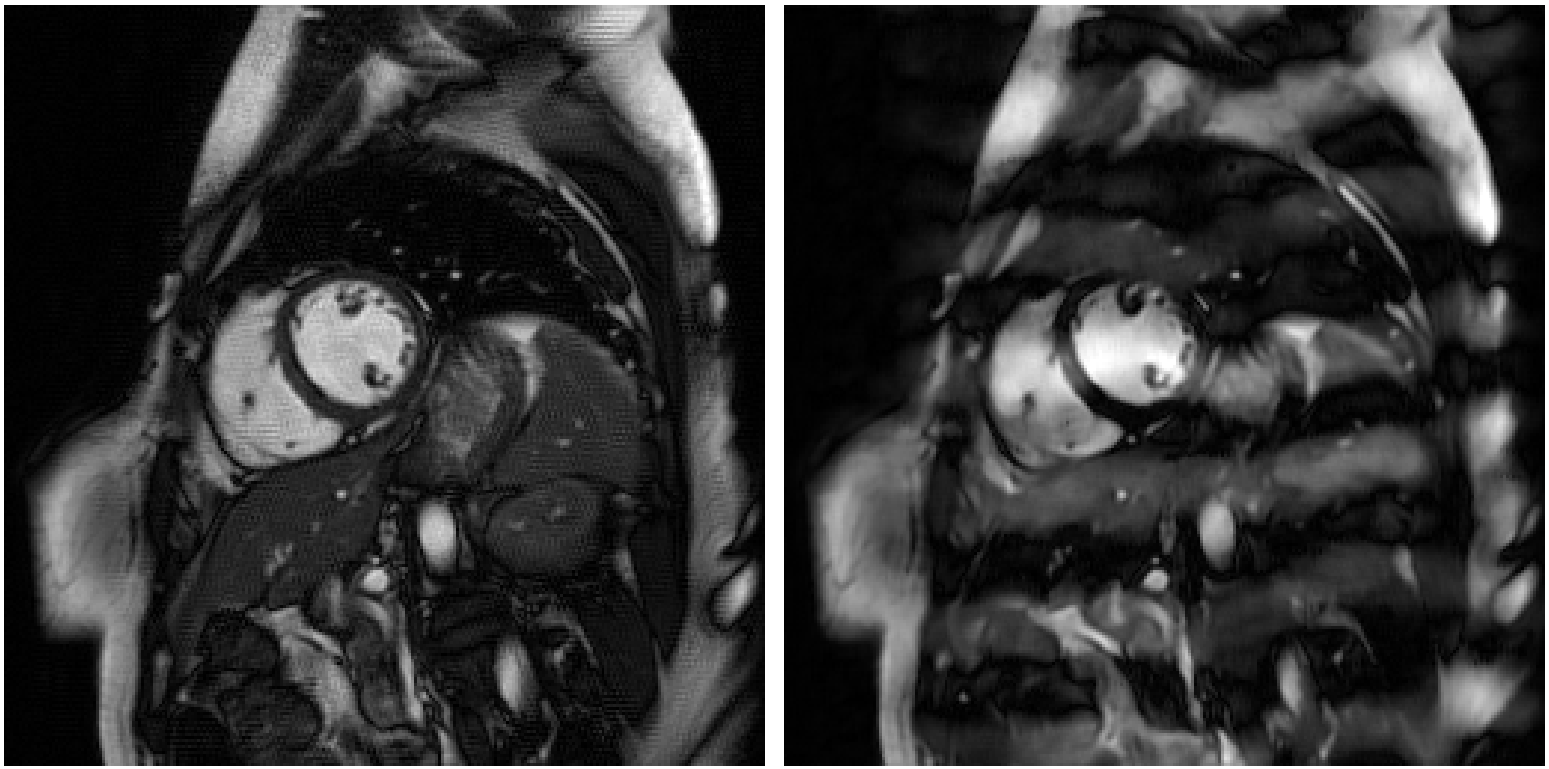
Deep MRI Net – Adversarial Noise



(a) $x + r$ (b) $f(A(x + r))$

Figure: Iterations: 2500

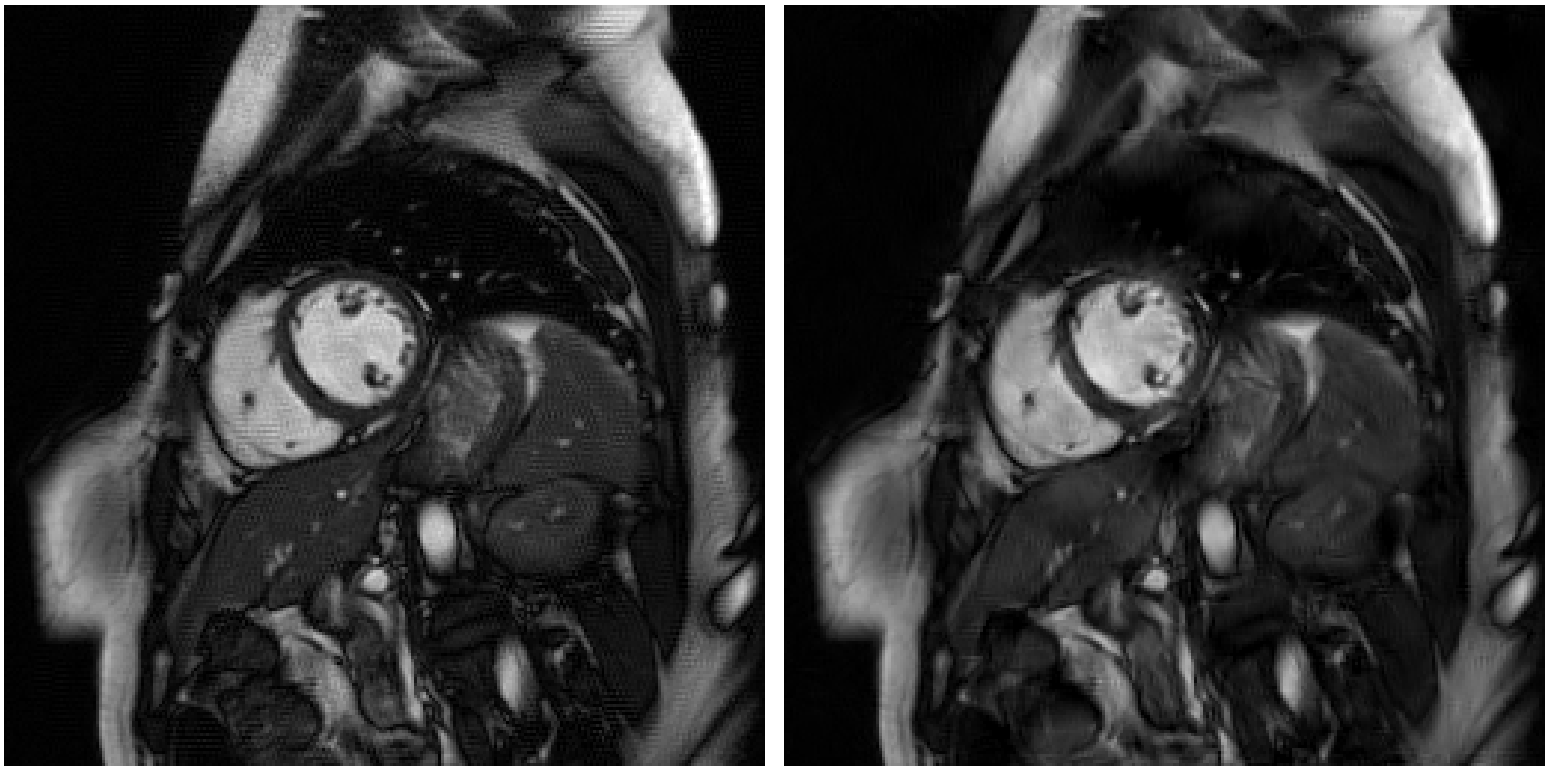
Deep MRI Net – Adversarial Noise



(a) $x + r$ (b) $f(A(x + r))$

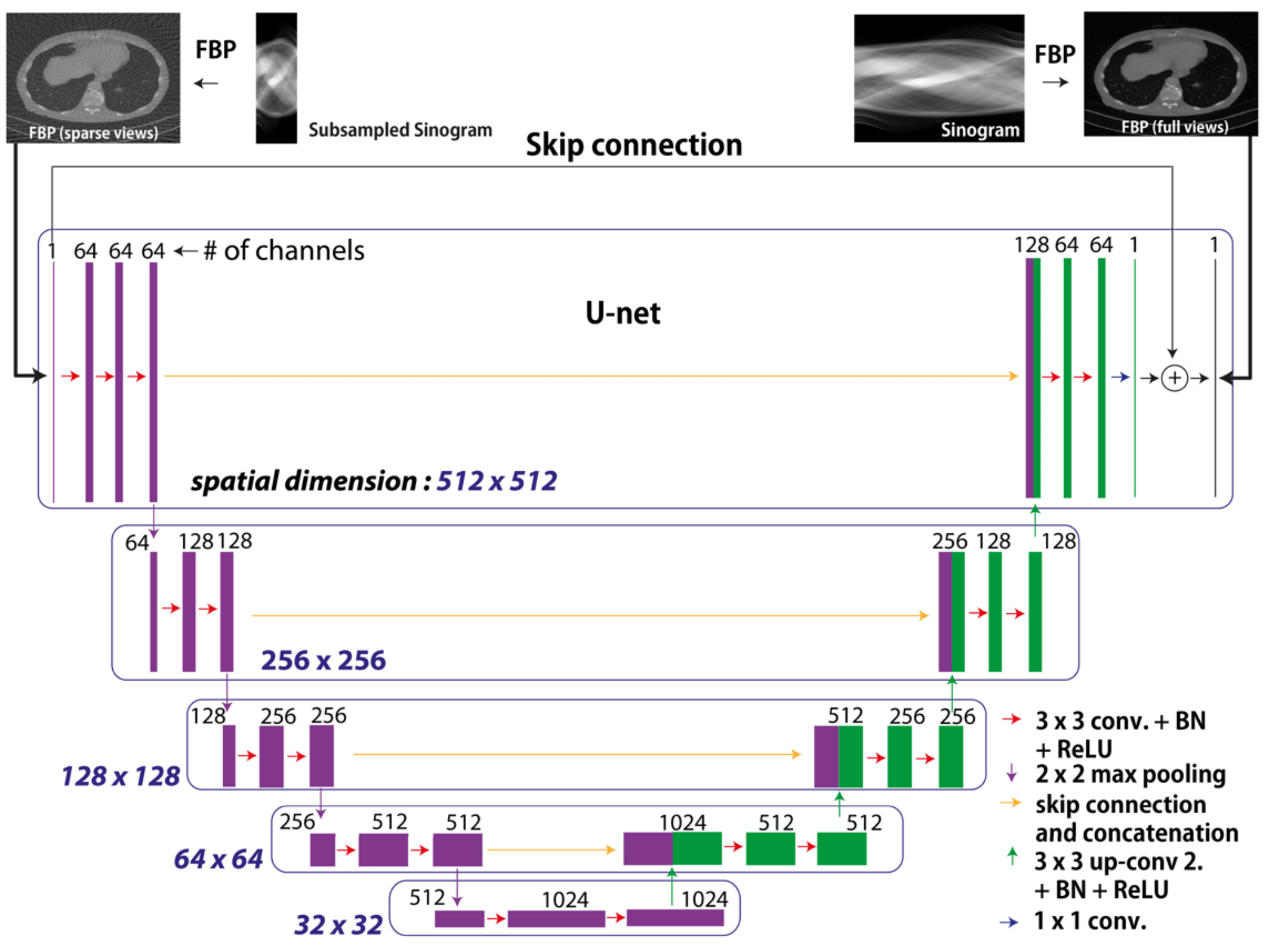
Figure: Iterations: 5000

Deep MRI Net – Adversarial Noise

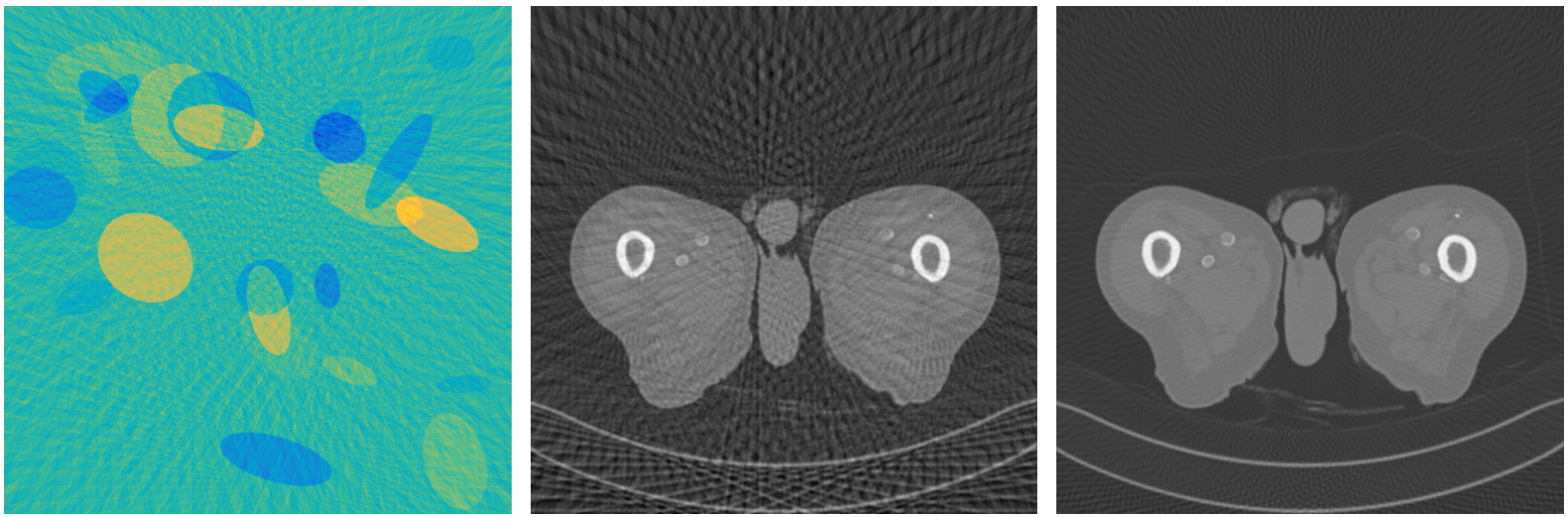


(a) $x + r$ (b) Compressive sensing

Figure: Iterations: 5000


FBP (sparse views)
FBP
Subsampled Sinogram
Sinogram
FBP
FBP (full views)
Skip connection
1 64 64 64 ← # of channels
U-net
128 64 64 1 1
spatial dimension : 512 x 512
64 128 128
256 128 128
256 x 256
128 256 256
512 256 256
128 x 128
256 512 512
1024 512 512
64 x 64
512 1024 1024
32 x 32
3 x 3 conv. + BN + ReLU
2 x 2 max pooling
skip connection and concatenation
3 x 3 up-conv 2. + BN + ReLU
1 x 1 conv.

FBPConvNet



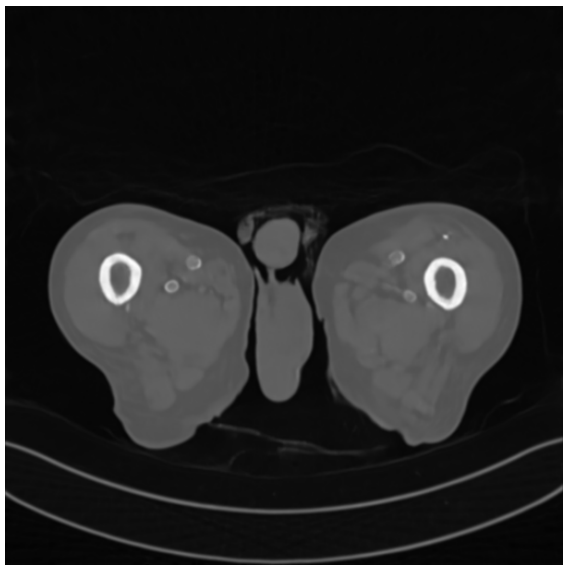
(a) Ell 50 (b) Med 50 (c) Med 143

Deep Convolutional Neural Network for Inverse Problems in Imaging (2016), K. Hwan Jin, M. T. McCann, E. Froustey, and M. Unser

FBPConvNet – Adversarial Noise

(a) x

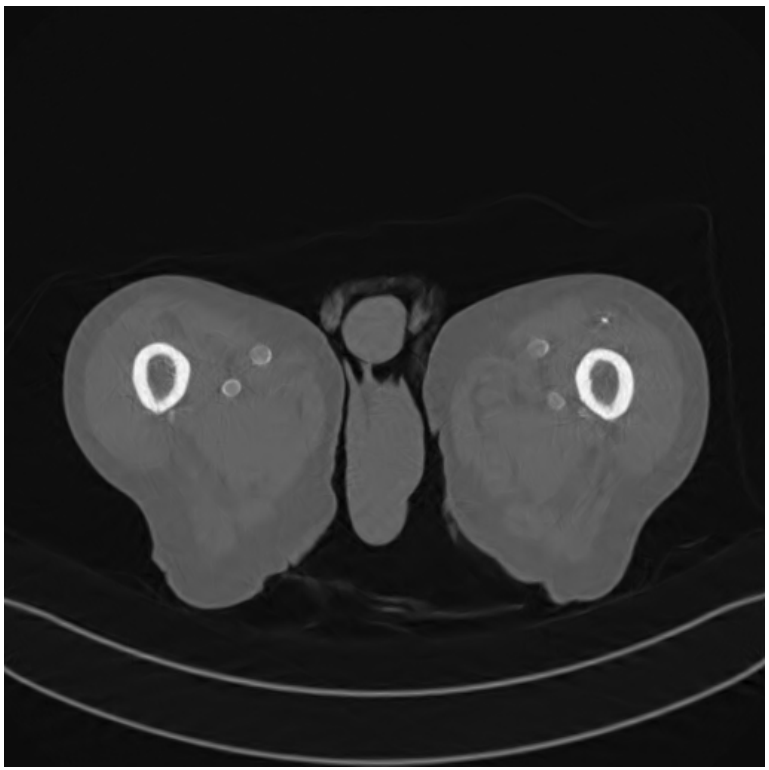
(b) $x + r$



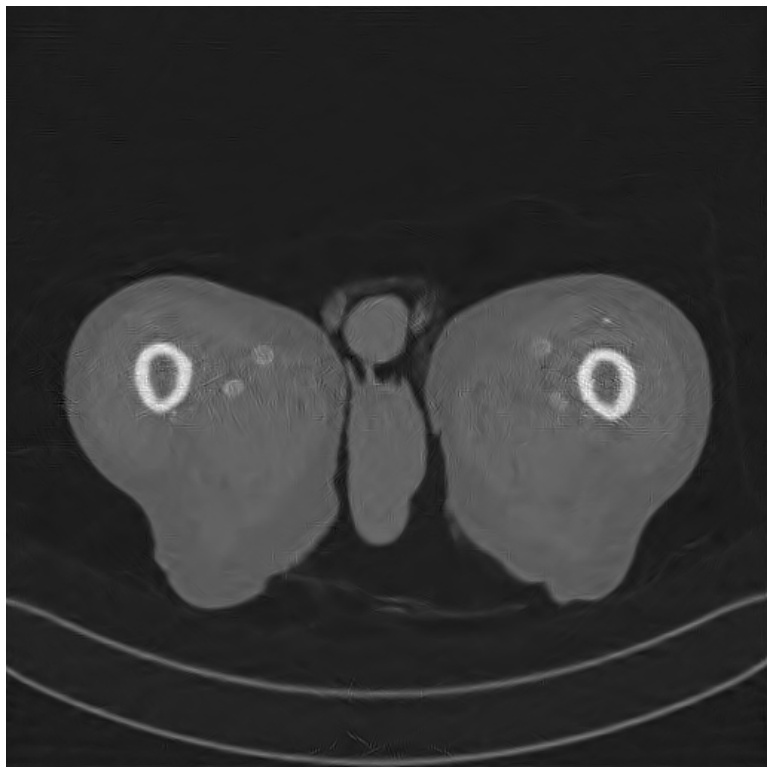
(c) $f(Ax)$

(d) $f(A(x + r))$

Compressive sensing reconstruction

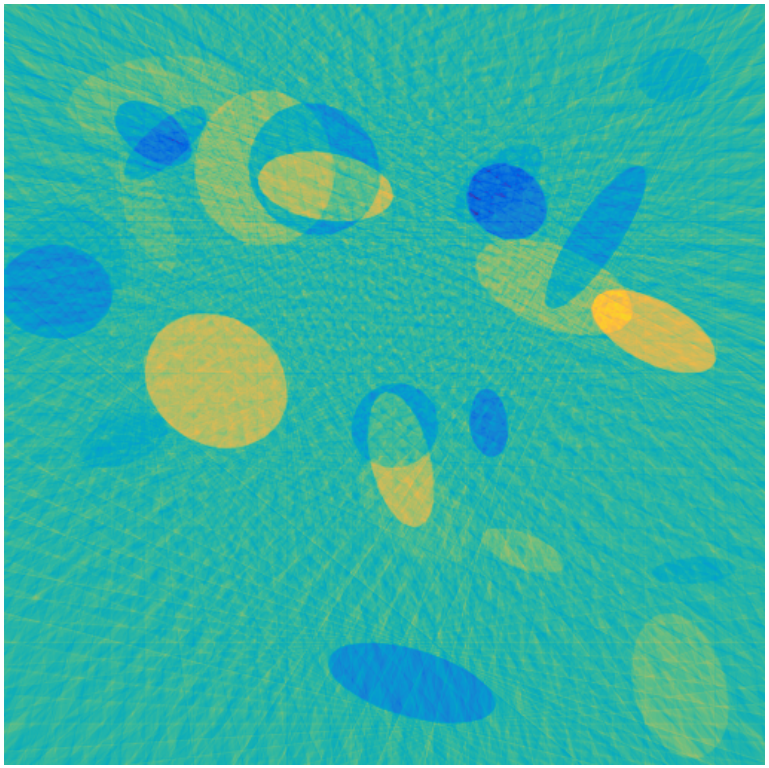


(a) Ax

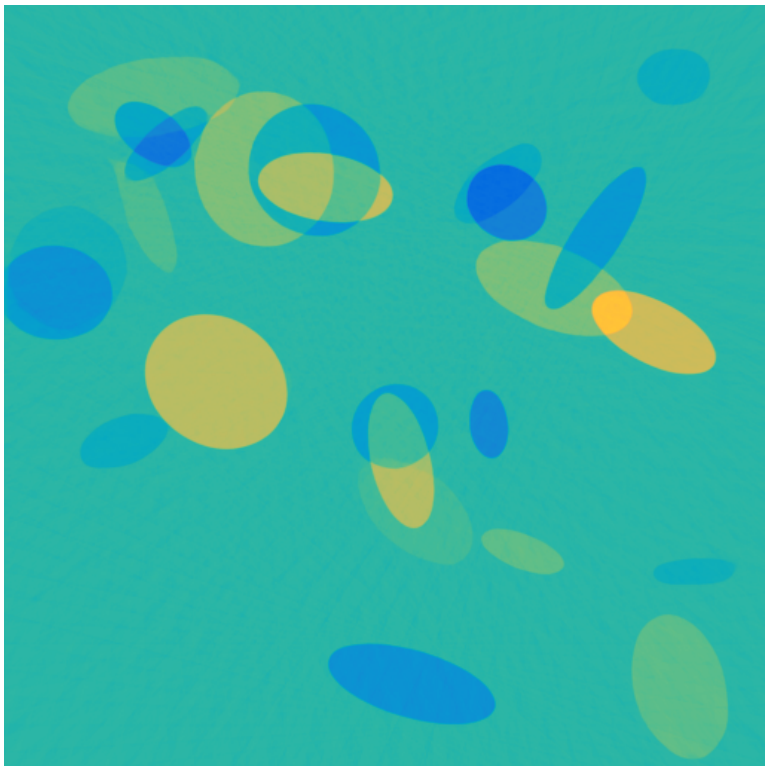


(b) $A(x + r)$

Adding more samples - 50 radial lines

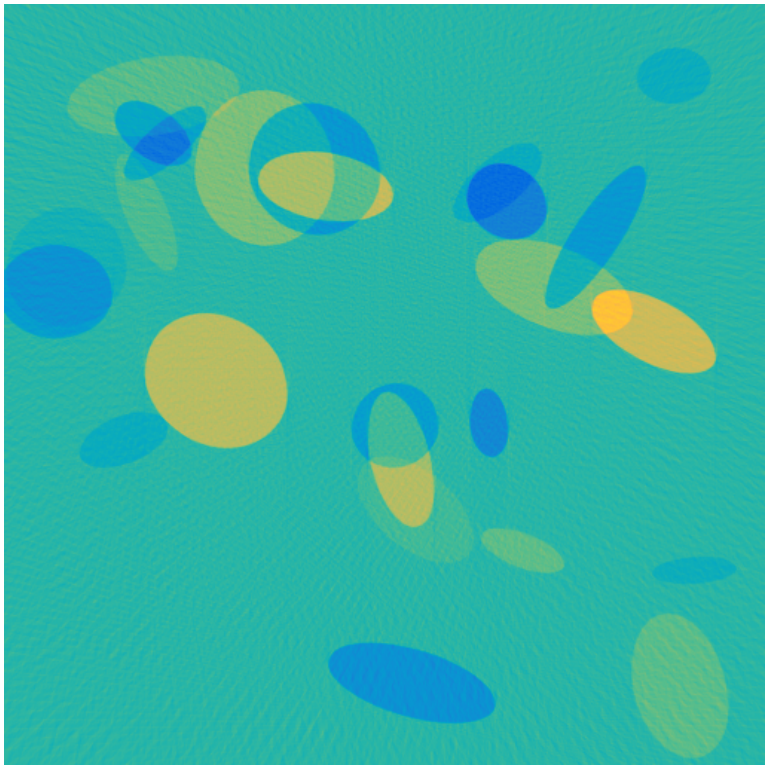


(a) $\tilde{A}^{-1}y$

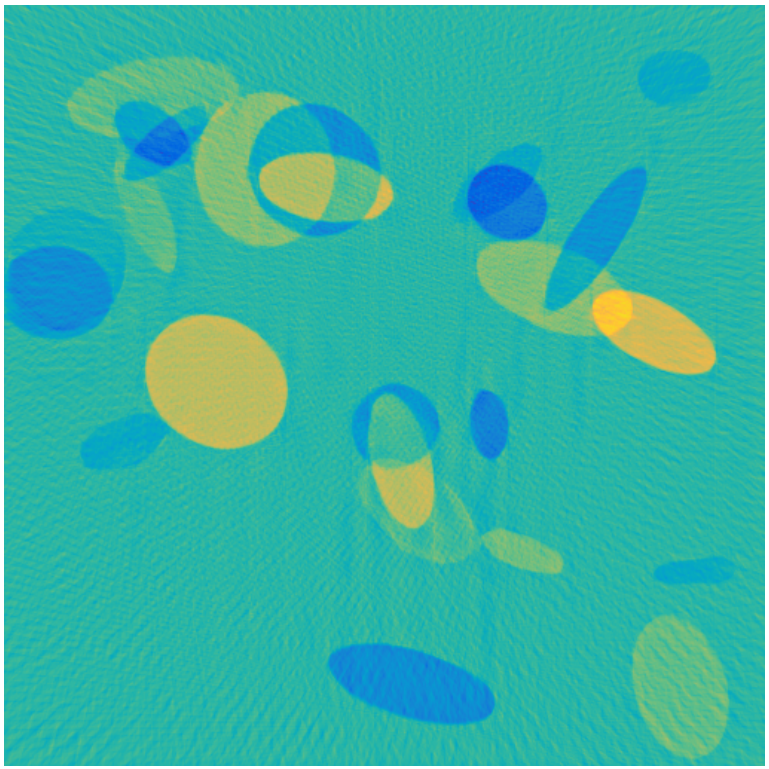


(b) $f(y)$

Adding more samples - 143 radial lines

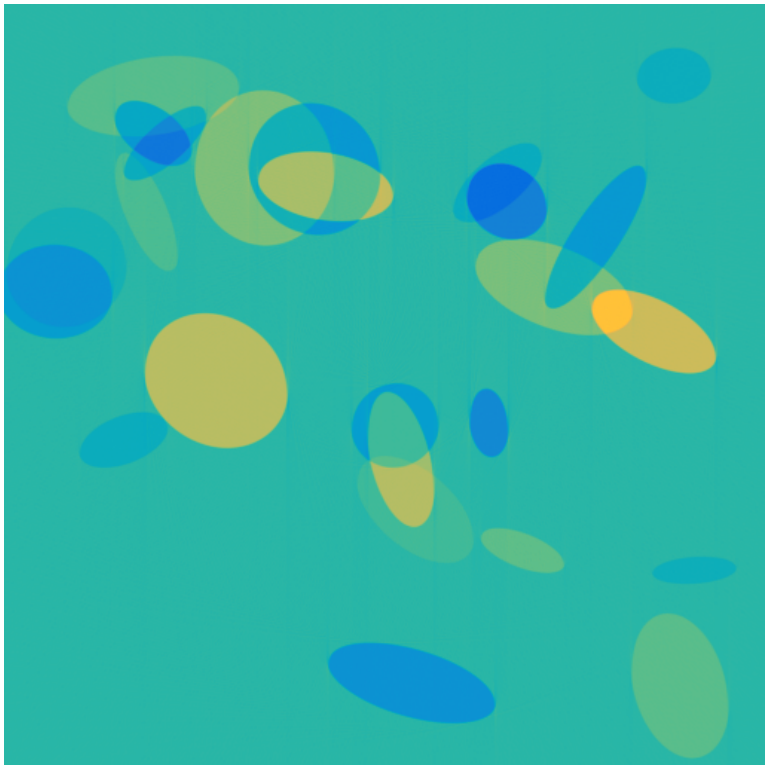


(a) $\tilde{A}^{-1}y$



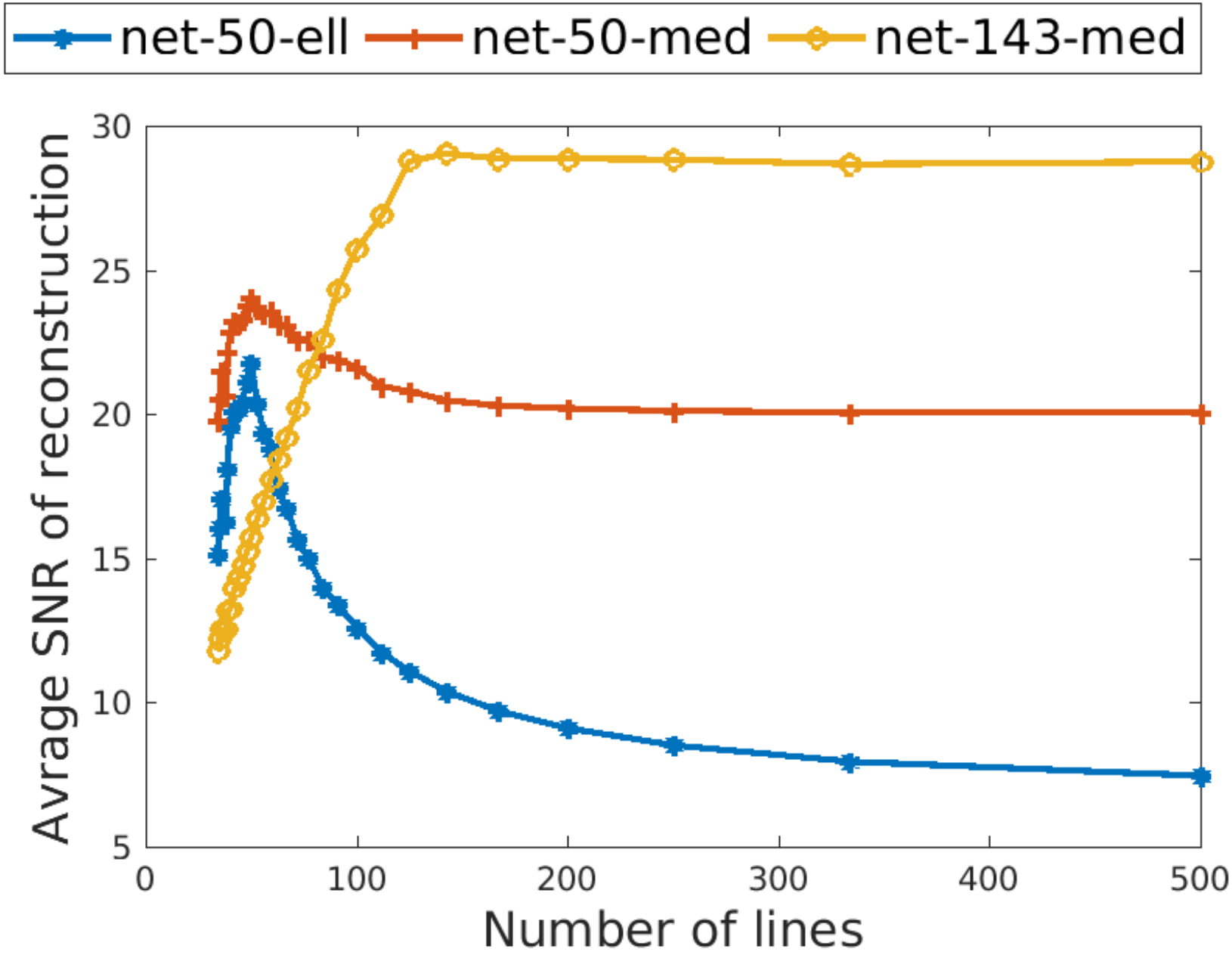
(b) $f(y)$

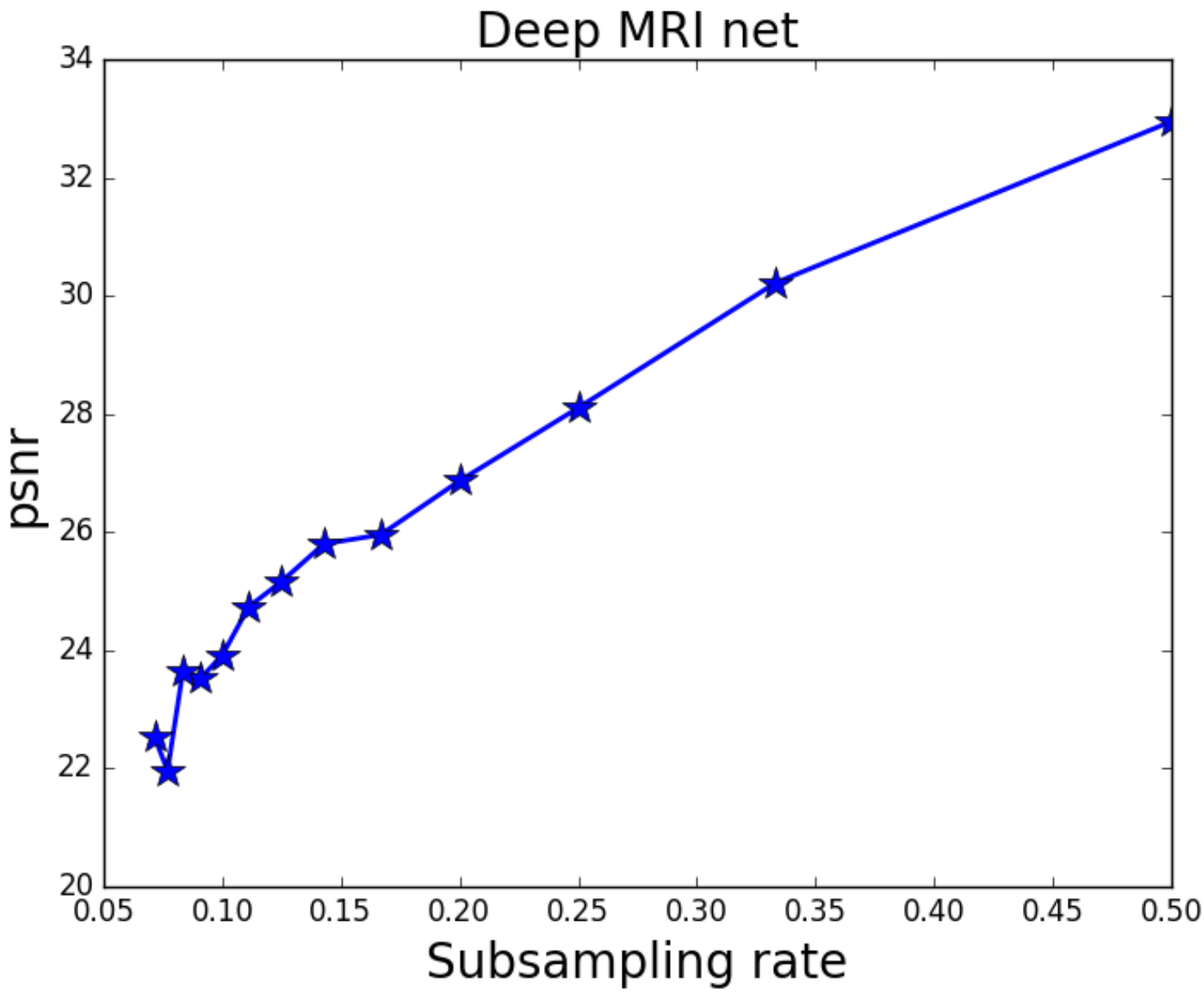
Adding more samples - 1000 radial lines



(a) $\tilde{A}^{-1}y$

(b) $f(y)$


net-50-ell
net-50-med
net-143-med
Avrage SNR of reconstruction
30
25
20
15
10
5
0
100
200
300
400
500
Number of lines

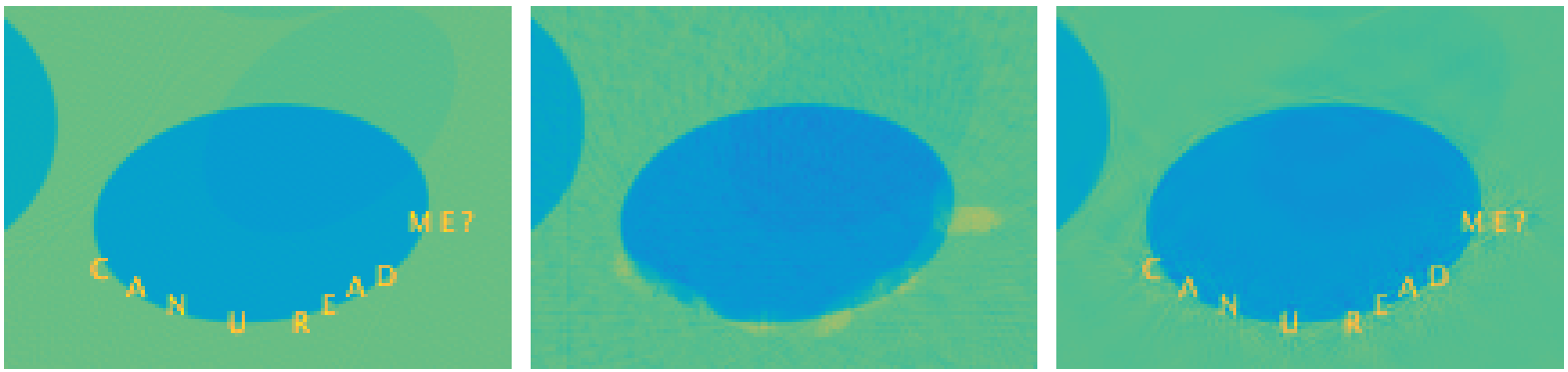

Deep MRI net
psnr
Subsampling rate
34
32
30
28
26
24
22
20
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0.50

Adding structure – FBPConvNet 50 radial lines

(a) x

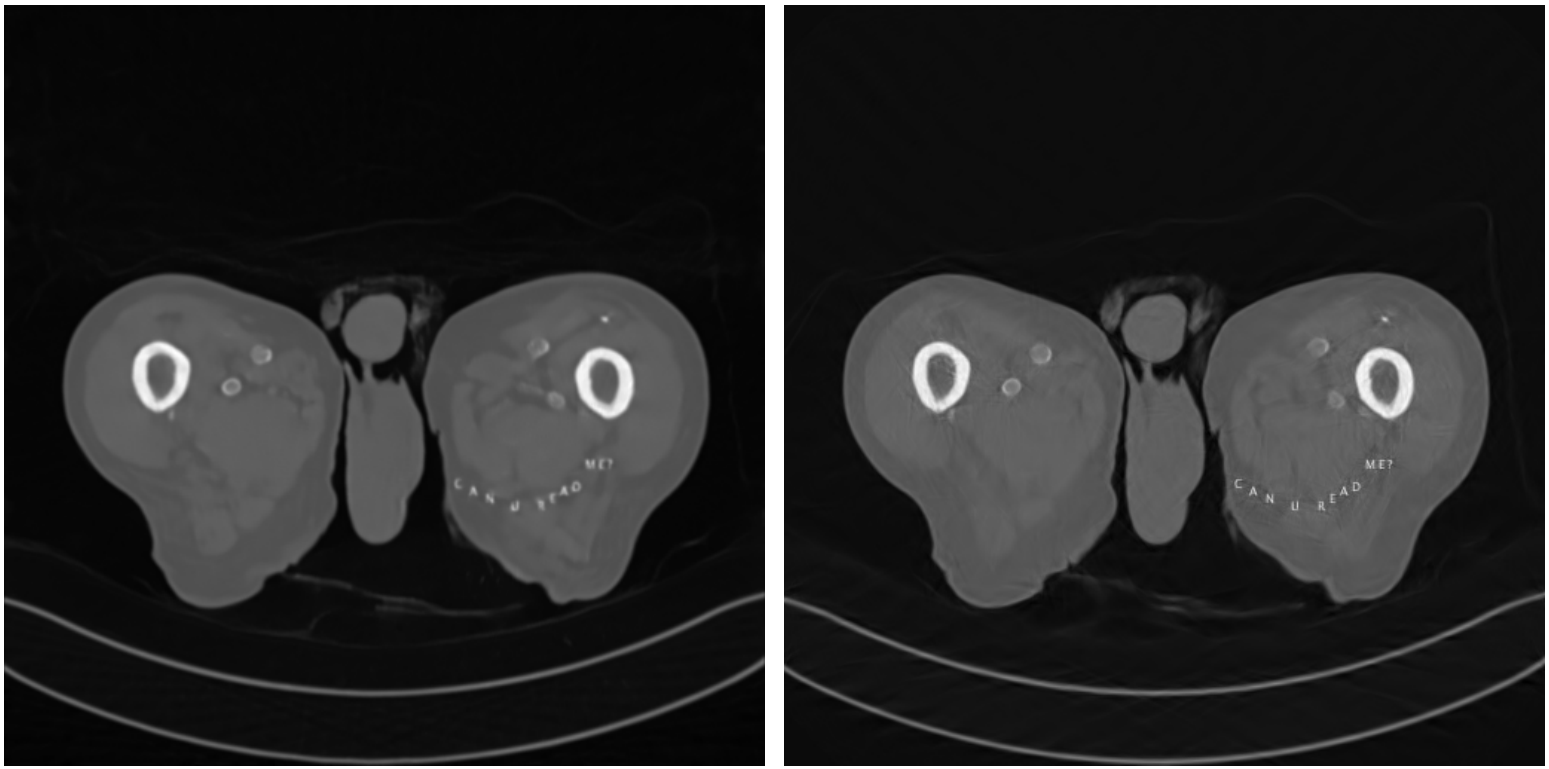
(b) $f(Ax)$

Adding structure – FBPConvNet 50 radial lines



(a) Original (b) Neural network (c) Comp. Sensing

Adding structure – FBPConvNet 50 radial lines



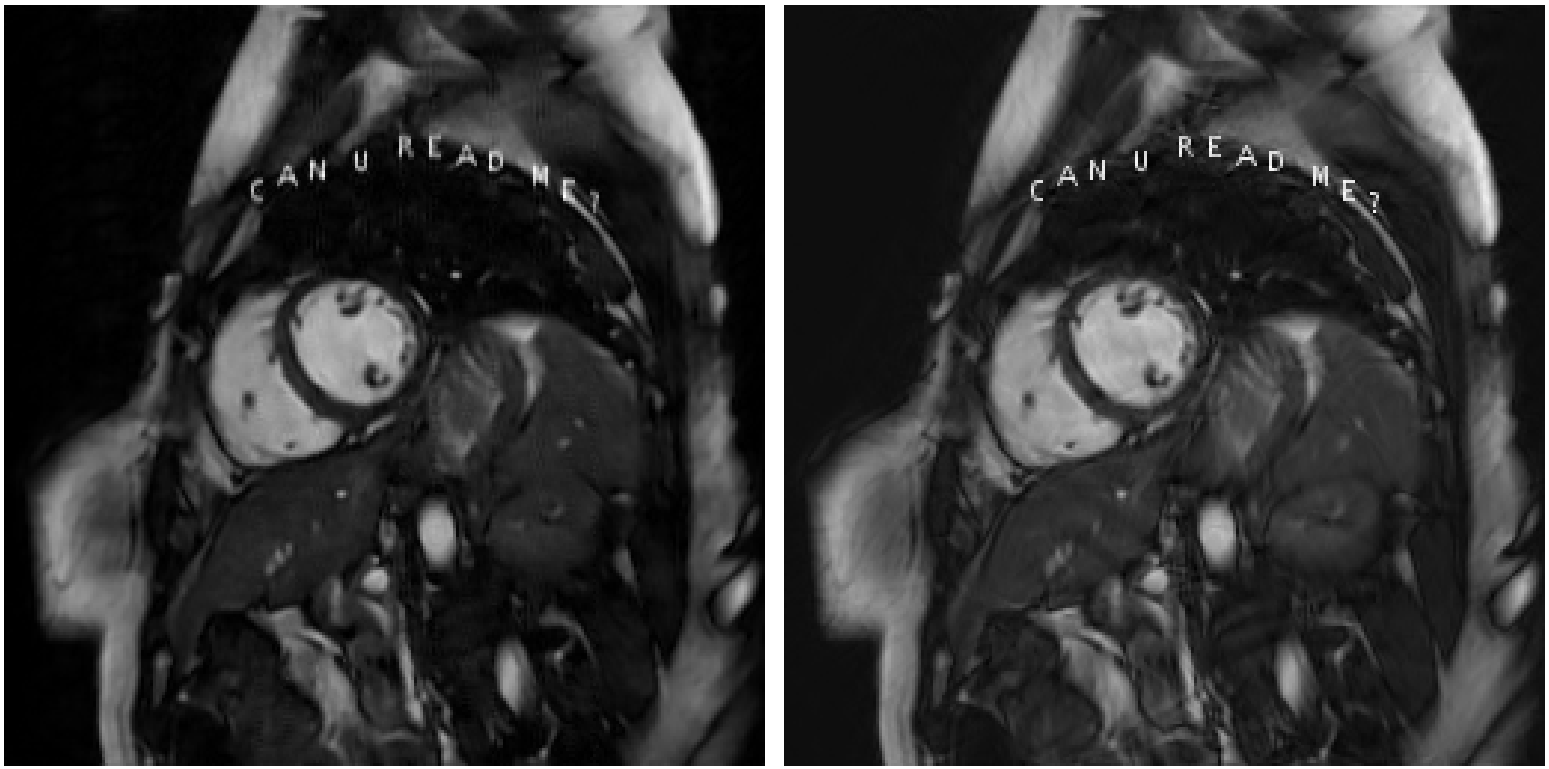
(a) Neural network (b) Compressive sensing

Adding structure – FBPConvNet 50 radial lines

(a) Neural network

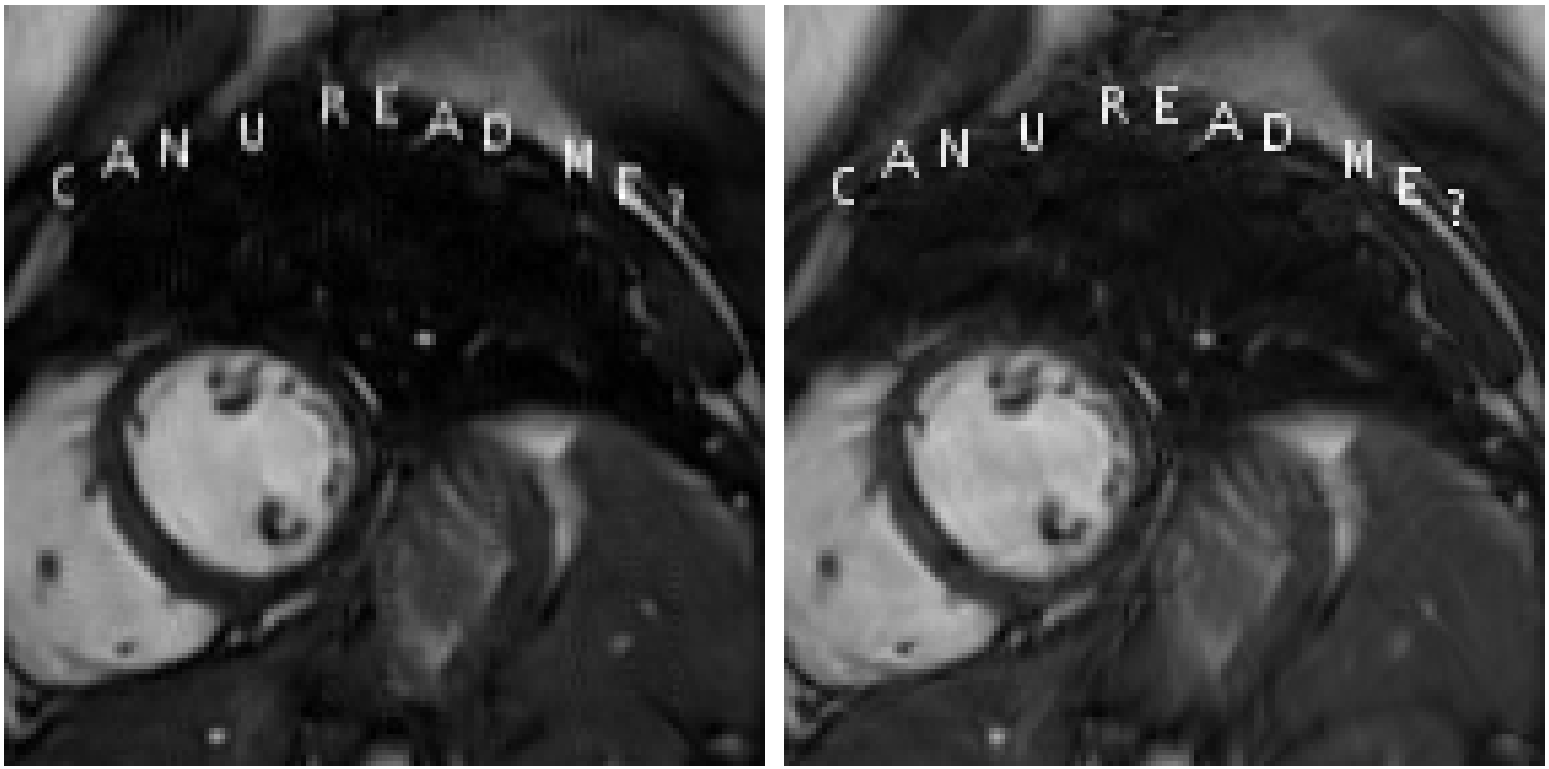
(b) Compressive sensing

Adding structure – Deep MRI Net



(a) Neural network (b) Compressive sensing

Adding structure – Deep MRI Net



(a) Neural network (b) Compressive sensing

Summary

Unstable with respect to ...

	ell 50	med 50	med 143	Deep MRI
Adversarial Noise	Yes	No	No	Yes
Adding more samples	Yes	Yes	No	No
Structural changes	Yes	Partly	No	No