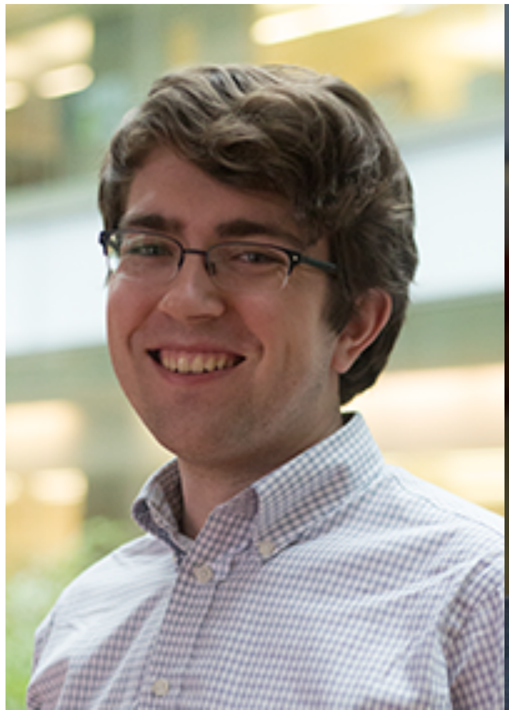
Machine Learning and Inverse Problems: Depth and Model Adaptation

Rebecca Willett, University of Chicago

Davis Gilton,
UW-Madison



Greg Ongie,
Marquette

Inverse problems in imaging

Observe: $y = Ax + \varepsilon$

Goal: Recover x from y

Classical approach: Tikhonov regularization (1943)

- Example: deblurring

- Least squares solution:

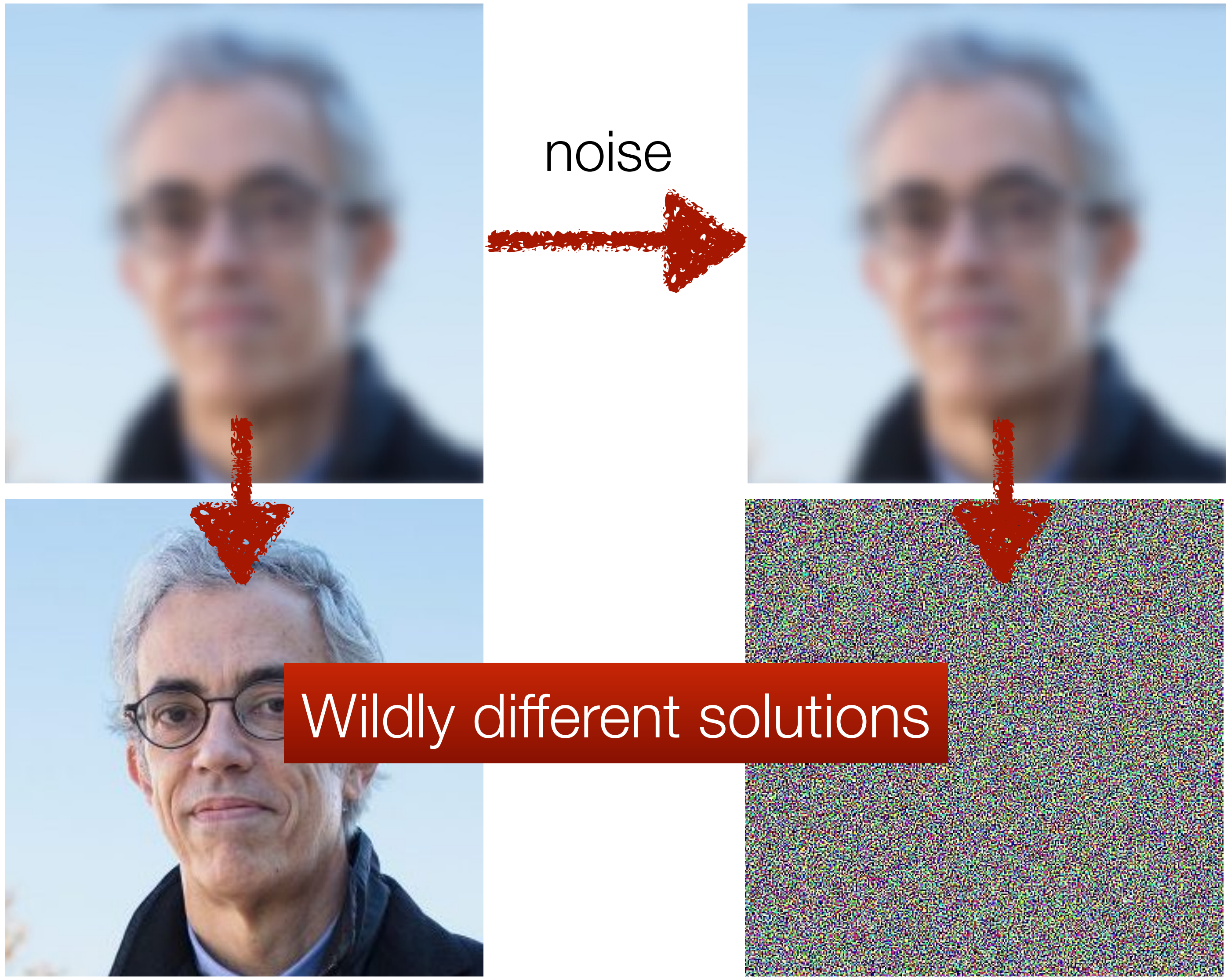
$$\hat{x} = (A^\top A)^{-1} A^\top y$$

$$= x + (A^\top A)^{-1} A^\top \varepsilon$$

Classical approach: Tikhonov regularization (1943)

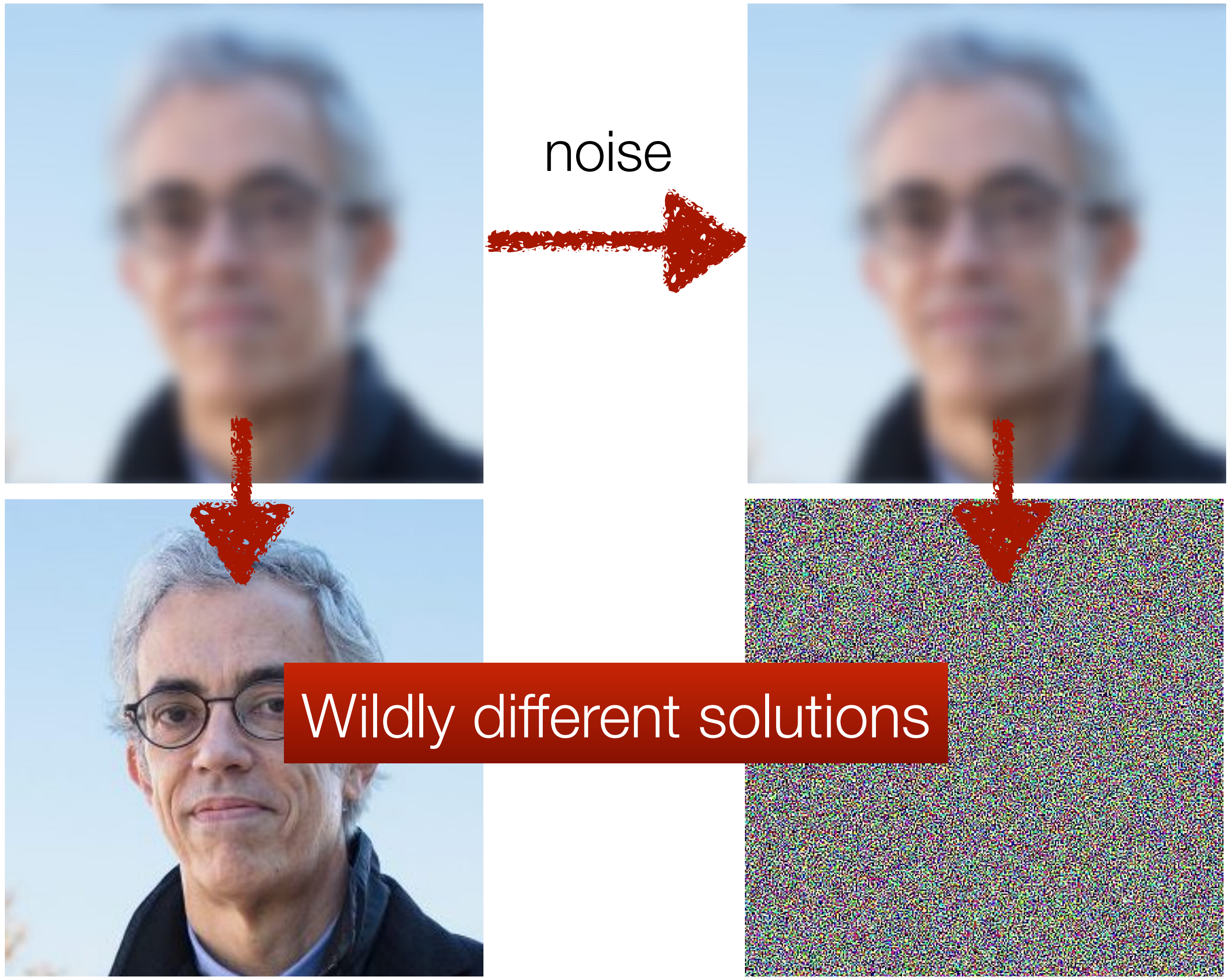
- Example: deblurring
- Least squares solution:

$$\hat{x} = (A^\top A)^{-1} A^\top y$$
$$= x + (A^\top A)^{-1} A^\top \varepsilon$$



Classical approach: Tikhonov regularization (1943)

- Example: deblurring



- Least squares solution:

$$\hat{x} = (A^\top A)^{-1} A^\top y$$
$$= x + (A^\top A)^{-1} A^\top \varepsilon$$

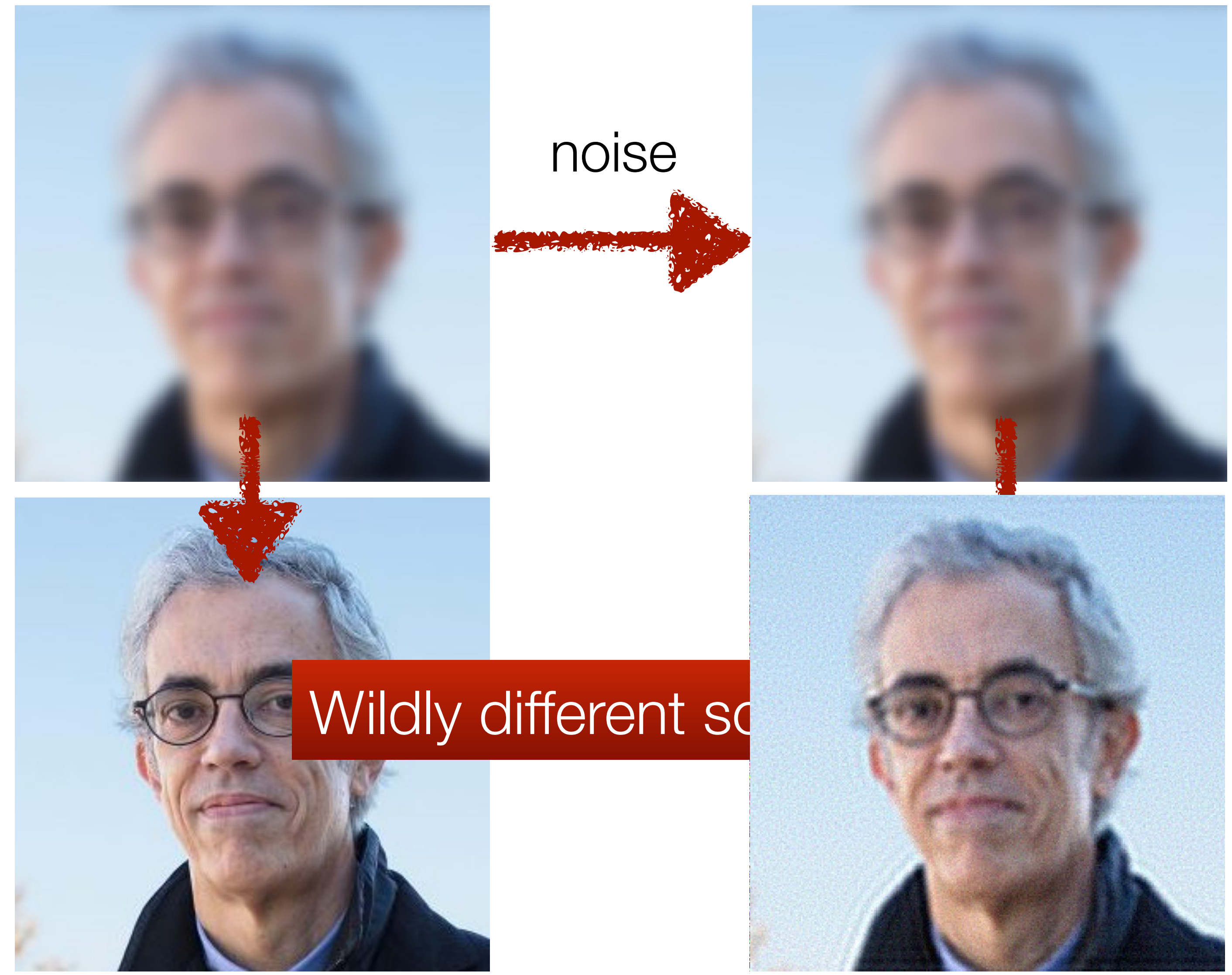
- Tikhonov regularization (aka "ridge regression")

$$\hat{x} = \arg\min_x \|y - Ax\|_2^2 + \lambda \|x\|_2^2$$
$$= (A^\top A + \lambda I)^{-1} A^\top y$$

better conditioned; suppresses noise

Classical approach: Tikhonov regularization (1943)

- Example: deblurring



Tikhonov regularization

- Least squares solution:

$$\hat{x} = (A^\top A)^{-1} A^\top y$$
$$= x + (A^\top A)^{-1} A^\top \varepsilon$$

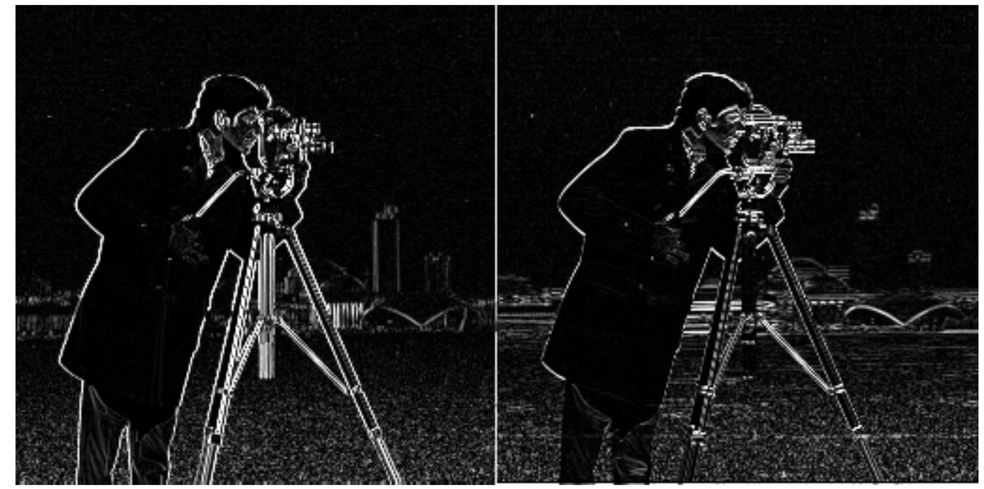
- Tikhonov regularization (aka "ridge regression")

$$\hat{x} = \arg\min_x \|y - Ax\|_2^2 + \lambda \|x\|_2^2$$
$$= (A^\top A + \lambda I)^{-1} A^\top y$$

better conditioned; suppresses noise

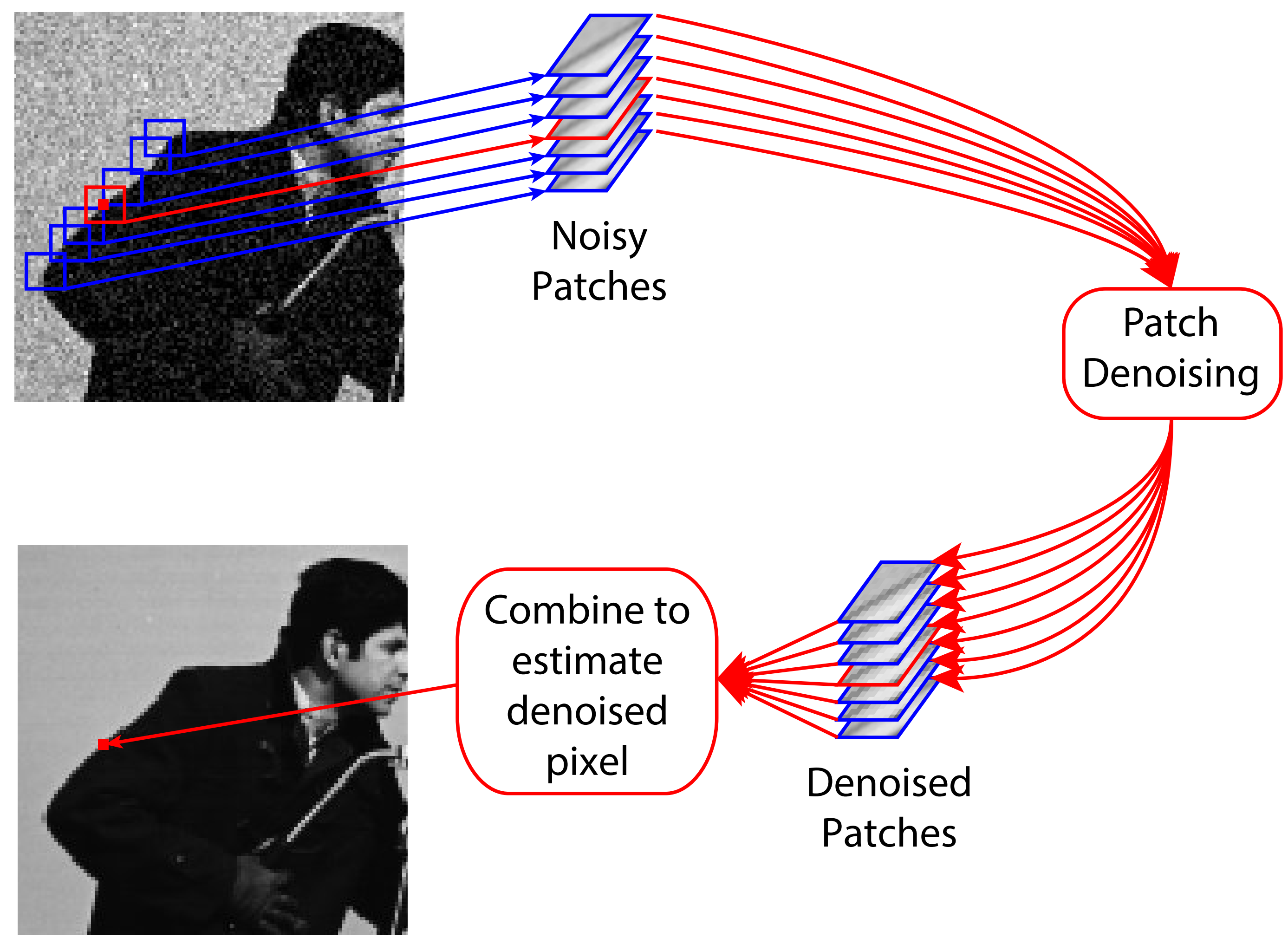
Geometric models of images

Total variation



(Wavelet) sparsity

Patch subspaces and manifolds



Geometric models reflect prior information about distribution of images

prior $-\log p(x) \iff R(x)$ regularizer

Learning to regularize

$$y \longrightarrow \boxed{\arg\min_x \|Ax - y\|^2 + R(x)} \longrightarrow \hat{x}$$

Instead of using choosing $R(x)$ a priori based on smoothness or geometric models, **can we learn $R(x)$ from training data?**

Arridge, Maass, Öktem, Schönlieb, 2019
Ongie, Jalal, Metzler, Baraniuk, Dimakis, Willett, 2020

Key tradeoffs

- **Generality vs. sample complexity:**
  Leveraging known A during training gives lower sample complexity, but model must be retrained for each new A
- **Training stability vs. convergence guarantees:**
  Unrolled methods with a small number of blocks (K) are easier to train but lack convergence guarantees
- **Reconstruction accuracy vs. sensitivity to model mismatch:**
  Learning to solve for A_0 may yield a regularizer ill-suited to $A_1 \neq A_0$

Key tradeoffs

- **Generality vs. sample complexity:**
Leveraging known A during training gives lower sample complexity, but model must be retrained for each new A
- **Training stability vs. convergence guarantees:**
Unrolled methods with a small number of blocks (K) are easier to train but lack convergence guarantees
- **Reconstruction accuracy vs. sensitivity to model mismatch:**
Learning to solve for A_0 may yield a regularizer ill-suited to $A_1 \neq A_0$

Example: Proximal Gradient Descent

$$y \longrightarrow \boxed{\arg\min_x \|Ax - y\|^2 + R(x)} \longrightarrow \hat{x}$$

set $x^{(0)}$ and step-size $\eta > 0$

for $k = 1, 2, \ldots$

$$z^{(k)} = x^{(k)} - \eta A^\top (Ax^{(k)} - y)$$ data consistency step

$$x^{(k+1)} = \arg\min_x \|x - z^{(k)}\|^2 + \eta R(x)$$ denoising step

Example: Proximal Gradient Descent

$$y \longrightarrow \boxed{\arg\min_x \|Ax - y\|^2 + R(x)} \longrightarrow \hat{x}$$

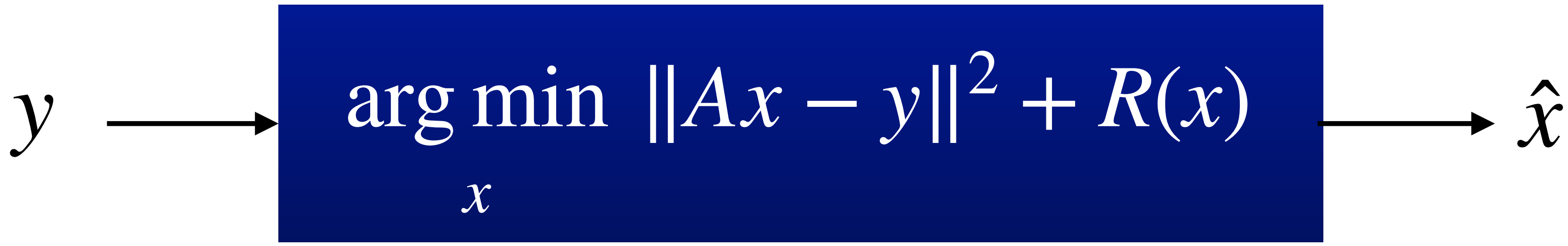
set $x^{(0)}$ and step-size $\eta > 0$

for $k = 1, 2, \ldots$

$$z^{(k)} = x^{(k)} - \eta A^\top (Ax^{(k)} - y)$$ data consistency step

$$x^{(k+1)} = \textbf{prox}_{\eta R}(z^{(k)})$$ denoising step

Example: Proximal Gradient Descent

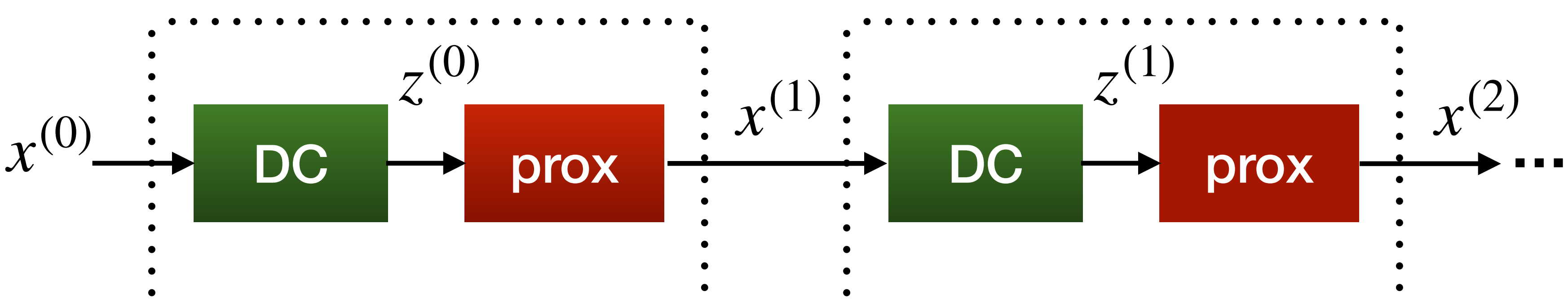


set $x^{(0)}$ and step-size $\eta > 0$

for $k = 1, 2, \ldots$

$$z^{(k)} = x^{(k)} - \eta A^\top (Ax^{(k)} - y)$$ data consistency step

$$x^{(k+1)} = \mathbf{prox}_{\eta R}(z^{(k)})$$ denoising step



repeat until convergence

Plug-and-Play Approach

$$y \longrightarrow \boxed{\arg\min_x \|Ax - y\|^2 + R(x)} \longrightarrow \hat{x}$$

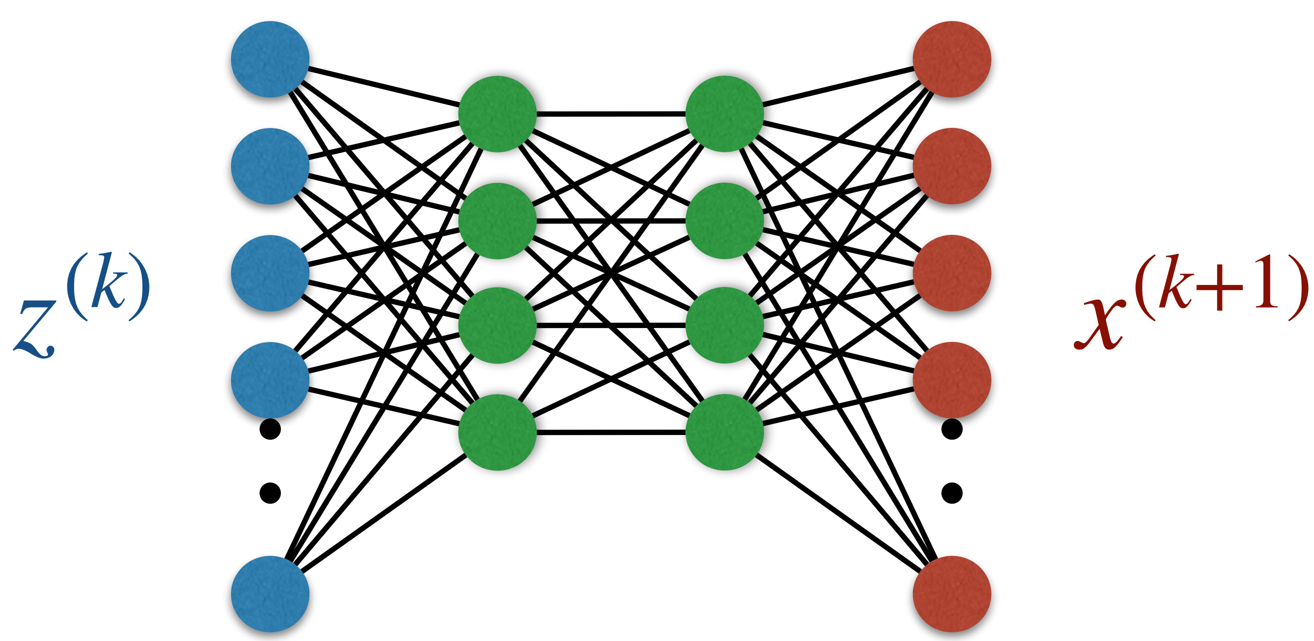
set $x^{(0)}$ and step-size $\eta > 0$

for $k = 1, 2, \ldots$

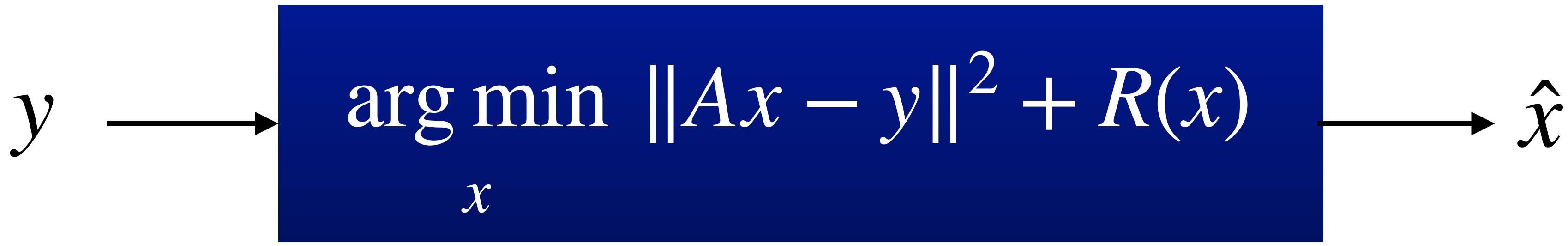
$$z^{(k)} = x^{(k)} - \eta A^\top (Ax^{(k)} - y)$$ data consistency step

$$x^{(k+1)} = \boxed{\textbf{prox}_{\eta R}(z^{(k)})}$$ denoising step

"Plug-in" a *pre-trained* CNN denoiser:



Plug-and-Play Approach



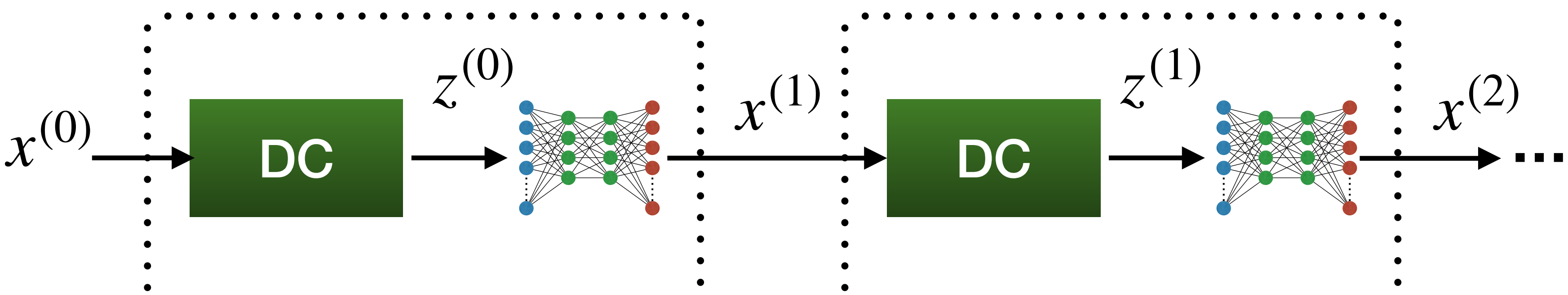
set $x^{(0)}$ and step-size $\eta > 0$

for $k = 1, 2, \ldots$

$$z^{(k)} = x^{(k)} - \eta A^\top (Ax^{(k)} - y)$$ data consistency step

$$x^{(k+1)} = \textbf{CNN}(z^{(k)})$$ denoising step

Plug-and-Play Prox-grad



repeat until convergence

- Plug-and-Play (Venkatakrishnan, Bouman, Wohlberg, 2013)
- Regularization-by-denoising (Romano, Elad, & Milanfar 2017)
- Convergence guarantees: (Ryu et al., 2019), (Reehorst & Schniter 2018)

How much training data?

Original
x

Observed
y

Reconstruction with convolutional neural network (CNN) trained with 80k samples

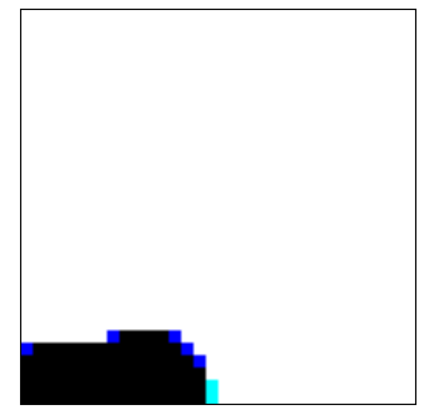
How much training data?

Original

x

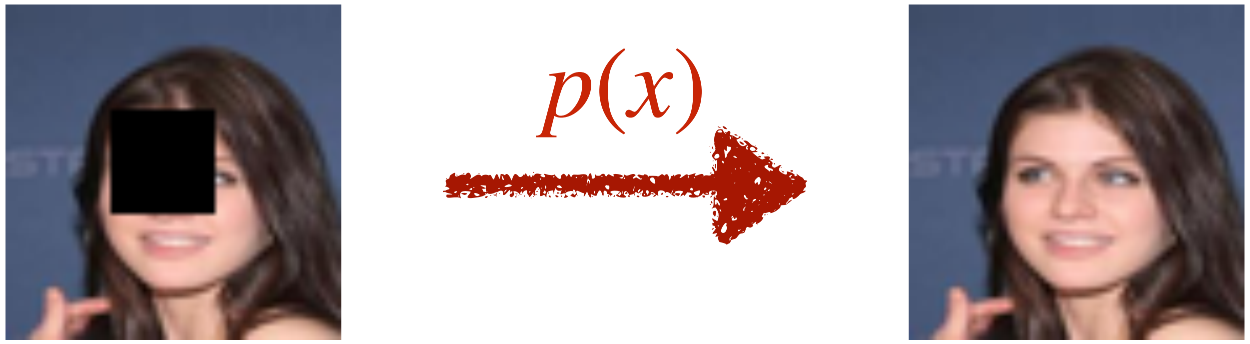
Observed

y

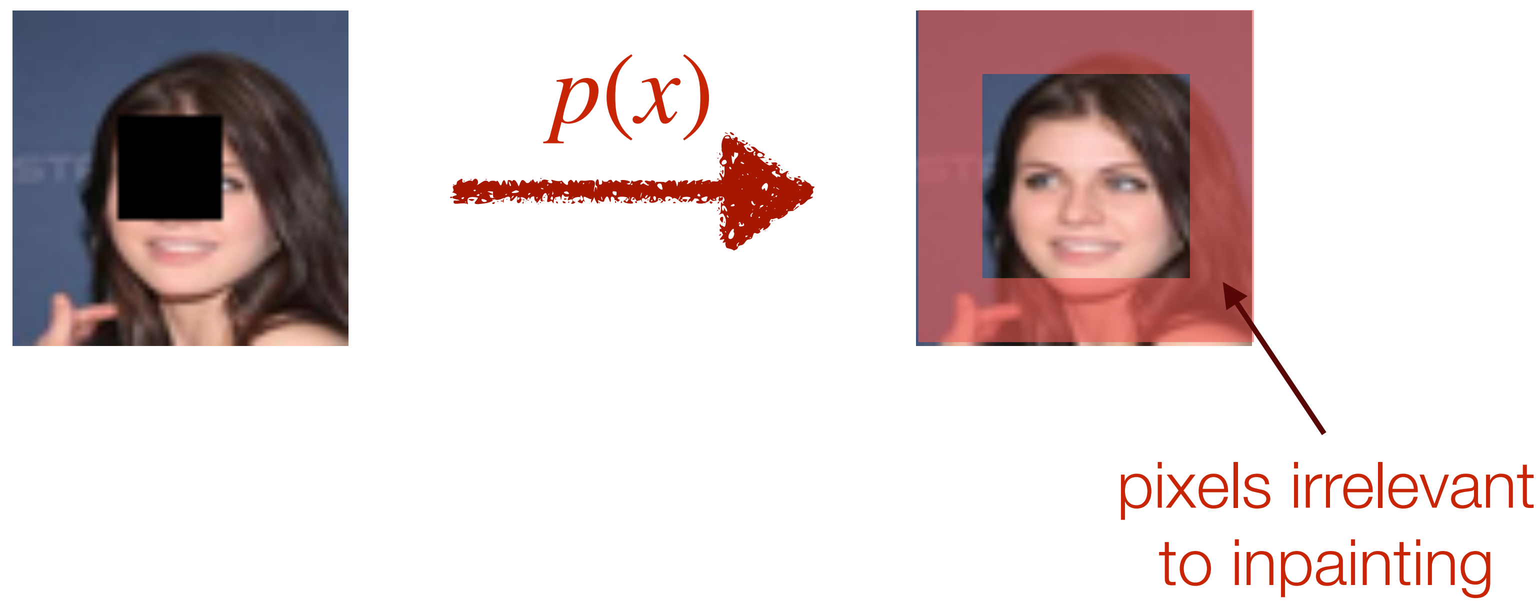


Reconstruction with convolutional neural network (CNN) trained with 2k samples

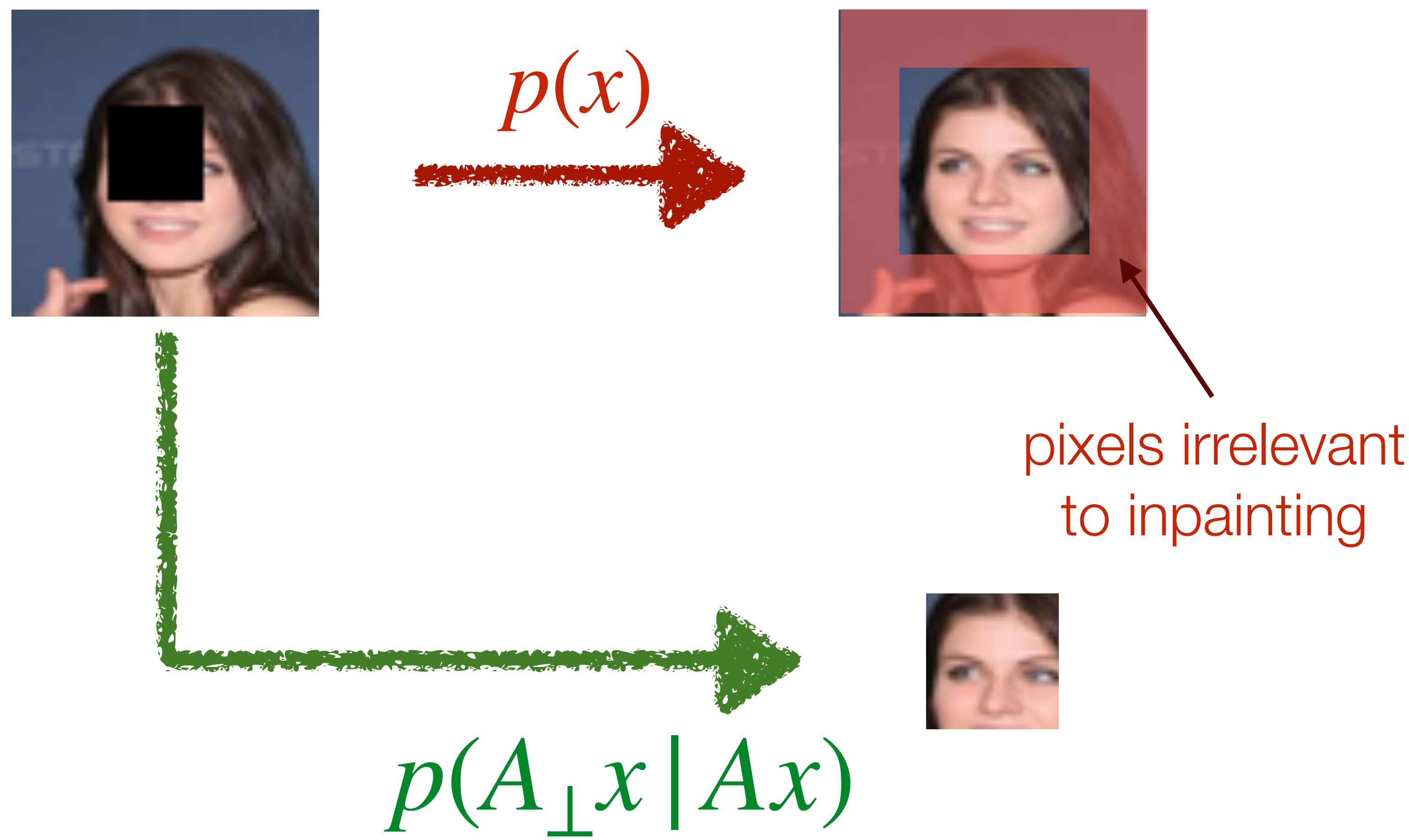
Prior vs. conditional density estimation



Prior vs. conditional density estimation

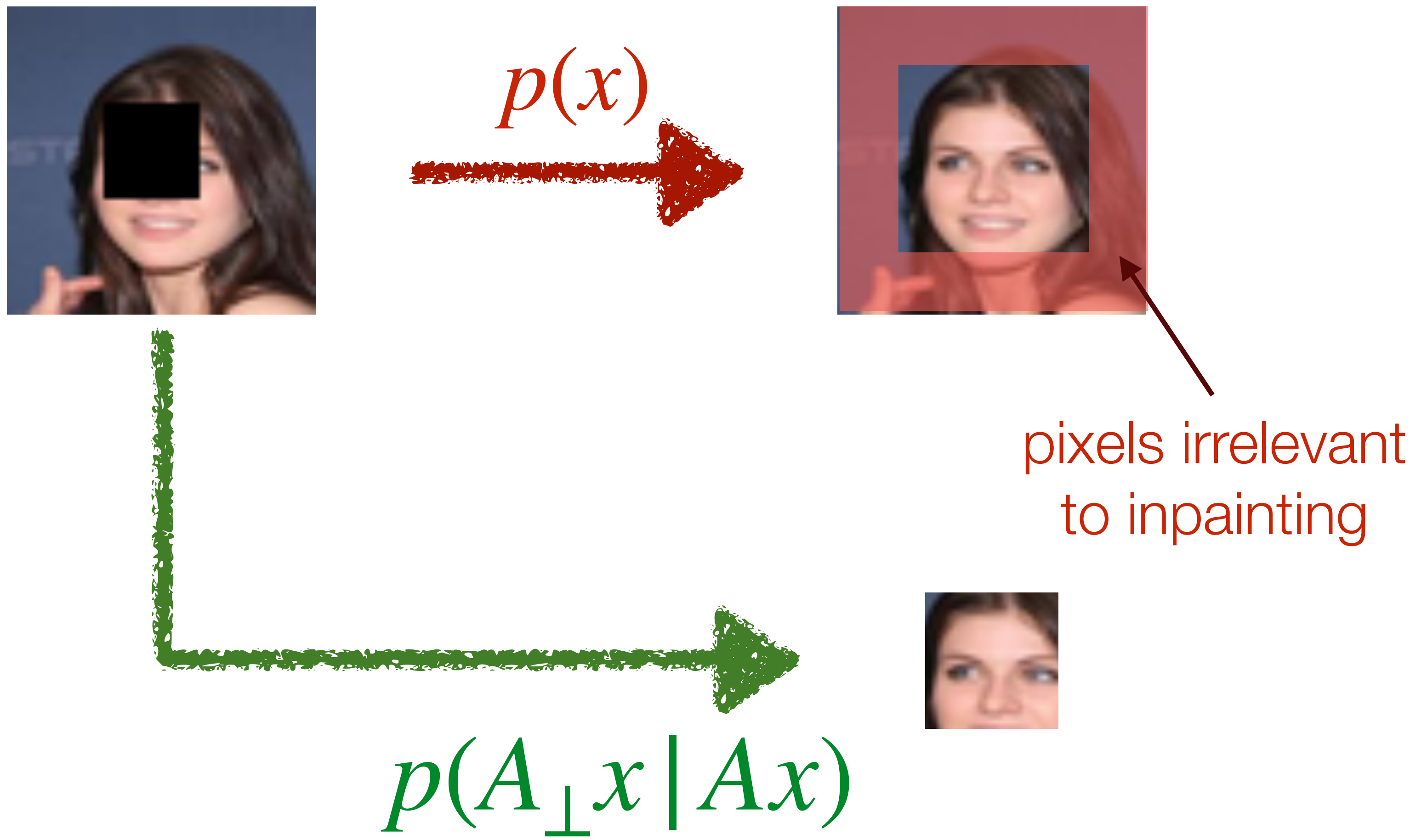


Prior vs. conditional density estimation



We need conditional density $p(A_{\perp}x \mid Ax)$

Prior vs. conditional density estimation



We need conditional density $p(A_\perp x \,|\, Ax)$

Estimating conditional density $p(A_\perp x \,|\, Ax)$ can require far fewer samples than estimating full density $p(x)$

Deep Unrolling

$$y \longrightarrow \boxed{\arg\min_x \|Ax - y\|^2 + R(x)} \longrightarrow \hat{x}$$

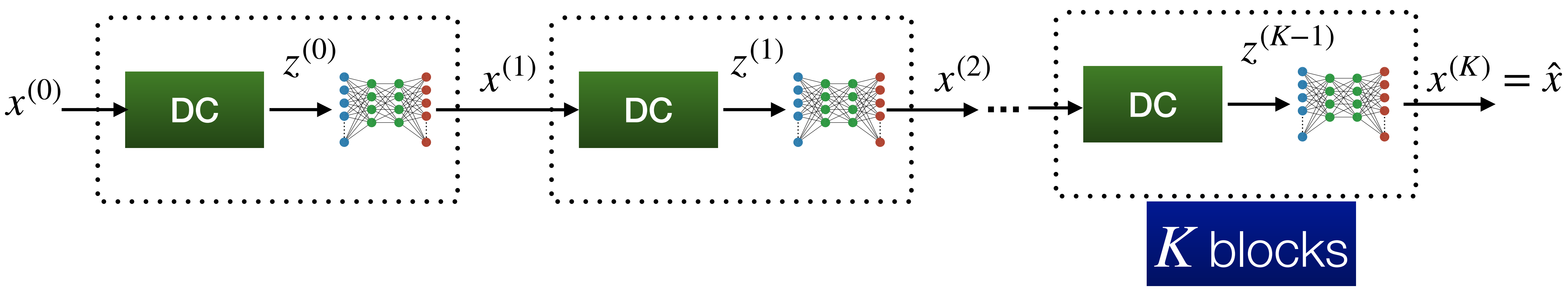
set $x^{(0)}$ and step-size $\eta > 0$

for $k = 1, 2, \ldots$

$$z^{(k)} = x^{(k)} - \eta A^\top (Ax^{(k)} - y)$$ data consistency step

$$x^{(k+1)} = \textbf{CNN}(z^{(k)})$$ denoising step

Deep Unrolling of Prox-grad



Deep Unrolling

$$y \longrightarrow \boxed{\arg\min_{x} \|Ax - y\|^2 + R(x)} \longrightarrow \hat{x}$$

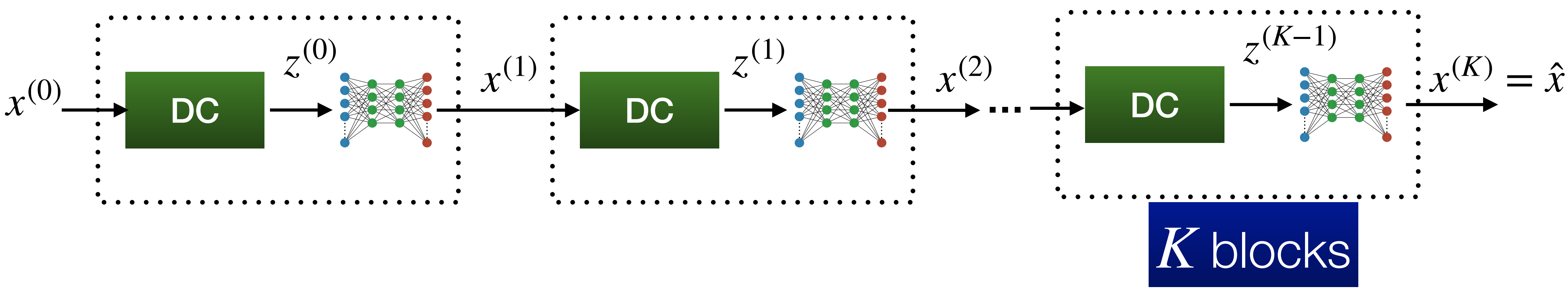
set $x^{(0)}$ and step-size $\eta > 0$

for $k = 1, 2, \ldots$

$$z^{(k)} = x^{(k)} - \eta A^\top (Ax^{(k)} - y)$$ data consistency step

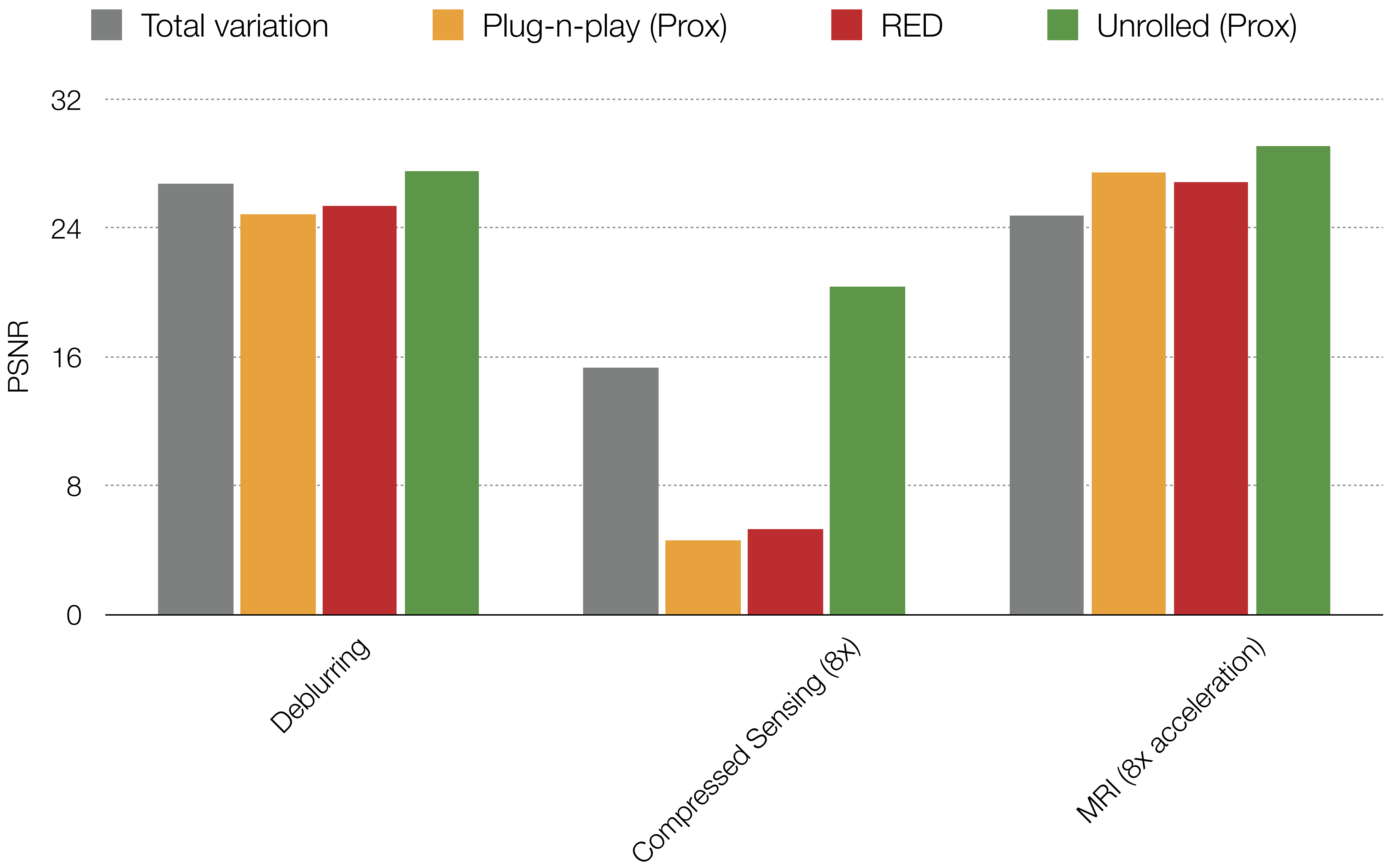
$$x^{(k+1)} = \textbf{CNN}(z^{(k)})$$ denoising step

Deep Unrolling of Prox-grad



"Unroll" K iterations, train end-to-end in a supervised manner using ground truth image/measurement pairs

Numerical results



Deep Unrolling

From: (Monga, Li, Eldar 2020)

TABLE I
SUMMARY OF RECENT METHODS EMPLOYING ALGORITHM UNROLLING IN PRACTICAL SIGNAL PROCESSING AND IMAGING APPLICATIONS.

Reference	Year	Application domain	Topics	Underlying Iterative Algorithms
Hershey *et al.* [30]	2014	Speech Processing	Signal channel source separation	Non-negative matrix factorization
Wang *et al.* [26]	2015	Computational imaging	Image super-resolution	Coupled sparse coding with iterative shrinkage and thresholding
Zheng *et al.* [31]	2015	Vision and Recognition	Semantic image segmentation	Conditional random field with mean-field iteration
Schuler *et al.* [32]	2016	Computational imaging	Blind image deblurring	Alternating minimization
Chen *et al.* [16]	2017	Computational imaging	Image denoising, JPEG deblocking	Nonlinear diffusion
Jin *et al.* [27]	2017	Medical Imaging	Sparse-view X-ray computed tomography	Iterative shrinkage and thresholding
Liu *et al.* [33]	2018	Vision and Recognition	Semantic image segmentation	Conditional random field with mean-field iteration
Solomon *et al.* [34]	2018	Medical imaging	Clutter suppression	Generalized ISTA for robust principal component analysis
Ding *et al.* [35]	2018	Computational imaging	Rain removal	Alternating direction method of multipliers
Wang *et al.* [36]	2018	Speech processing	Source separation	Multiple input spectrogram inversion
Adler *et al.* [37]	2018	Medical Imaging	Computational tomography	Proximal dual hybrid gradient
Wu *et al.* [38]	2018	Medical Imaging	Lung nodule detection	Proximal dual hybrid gradient
Yang *et al.* [14]	2019	Medical imaging	Medical resonance imaging, compressive imaging	Alternating direction method of multipliers
Hosseini *et al.* [39]	2019	Medical imaging	Medical resonance imaging	Proximal gradient descent
Li *et al.* [40]	2019	Computational imaging	Blind image deblurring	Half quadratic splitting
Zhang *et al.* [41]	2019	Smart power grids	Power system state estimation and forecasting	Double-loop prox-linear iterations
Zhang *et al.* [42]	2019	Computational imaging	Blind image denoising, JPEG deblocking	Moving endpoint control problem
Lohit *et al.* [43]	2019	Remote sensing	Multi-spectral image fusion	Projected gradient descent
Yoffe *et al.* [44]	2020	Medical Imaging	Super resolution microscopy	Sparsity-based super-resolution microscopy from correlation information [45]

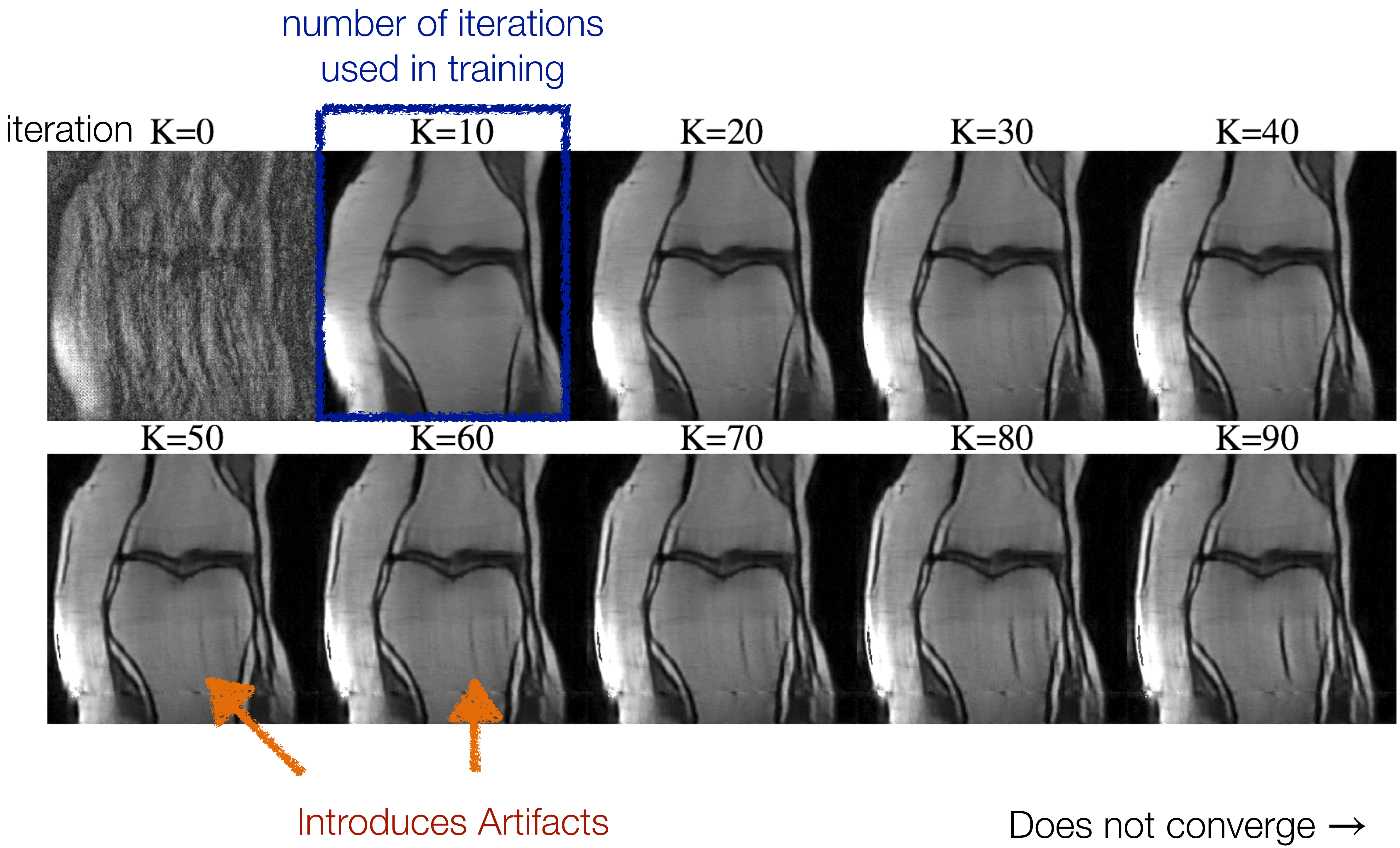
Key tradeoffs

- **Generality vs. sample complexity:**
  Leveraging known A during training gives lower sample complexity, but model must be retrained for each new A
- **Training stability vs. convergence guarantees:**
  Unrolled methods with a small number of blocks (K) are easier to train but lack convergence guarantees
- **Reconstruction accuracy vs. sensitivity to model mismatch:**
  Learning to solve for A_0 may yield a regularizer ill-suited to $A_1 \neq A_0$

Key tradeoffs

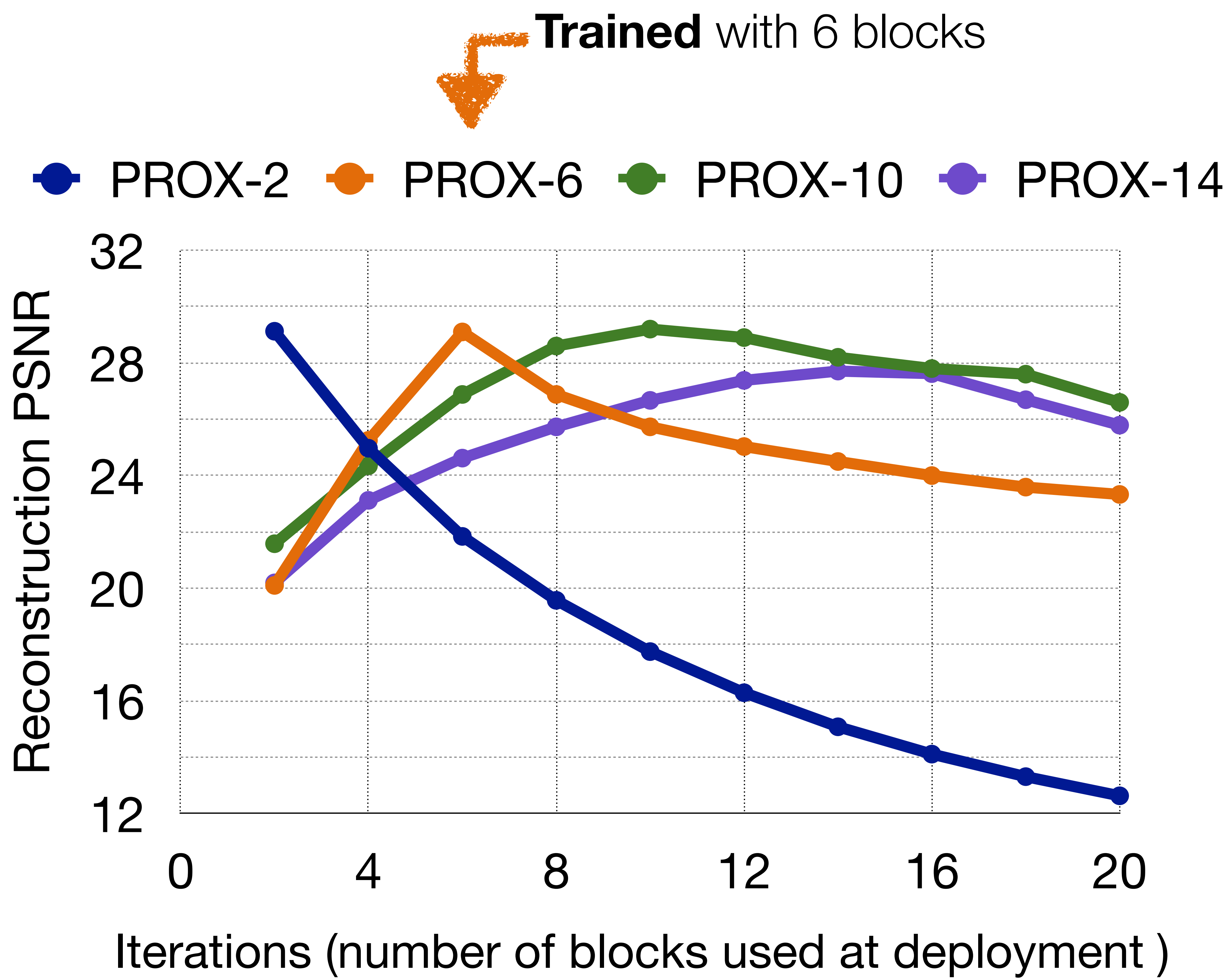
- **Generality vs. sample complexity:**
  Leveraging known A during training gives lower sample complexity, but model must be retrained for each new A
- **Training stability vs. convergence guarantees:**
  Unrolled methods with a small number of blocks (K) are easier to train but lack convergence guarantees
- **Reconstruction accuracy vs. sensitivity to model mismatch:**
  Learning to solve for A_0 may yield a regularizer ill-suited to $A_1 \neq A_0$

Deep Unrolling - Are we really learning a prox/denoiser?



Gilton, Ongie, & Willett 2021

Deep Unrolling and Iterations



Plug-and-play

- No new training required
- Convergence guarantees
- Outperformed by end-to-end learning

Deep Unrolling

- Demanding to train
- No convergence guarantees
- State-of-the-art results

Deep Equilibrium (Proposed)

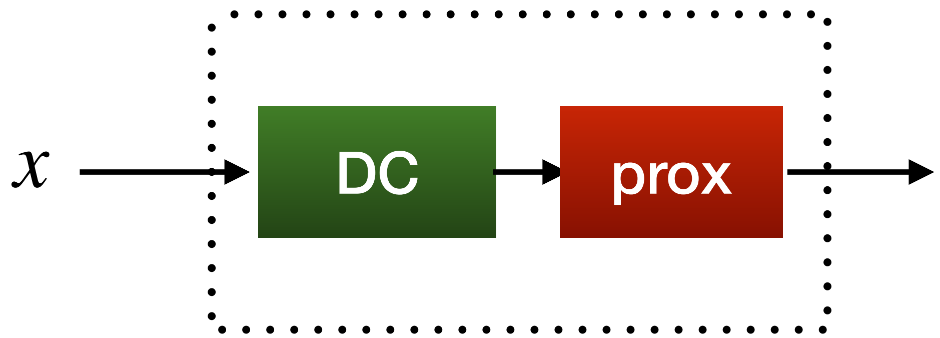
- Lightweight to train
- Convergence guarantees
- State-of-the-art results

Deep Equilibrium Models

Most iterative reconstruction algorithms can be interpreted as solving for a **fixed-point** of a non-linear operator $f(\,\cdot\,)$

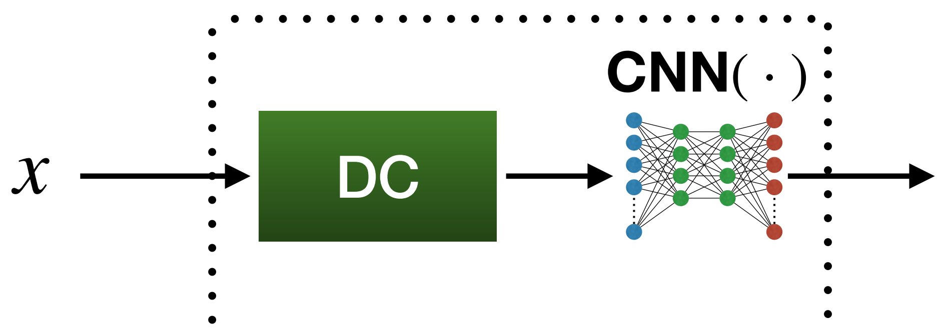
$$\text{find } x^* \text{ such that } x^* = f(x^*)$$

ex: proximal gradient descent



$$f(x) = \textbf{prox}_{\eta R}(x - \eta A^\top(Ax - y))$$

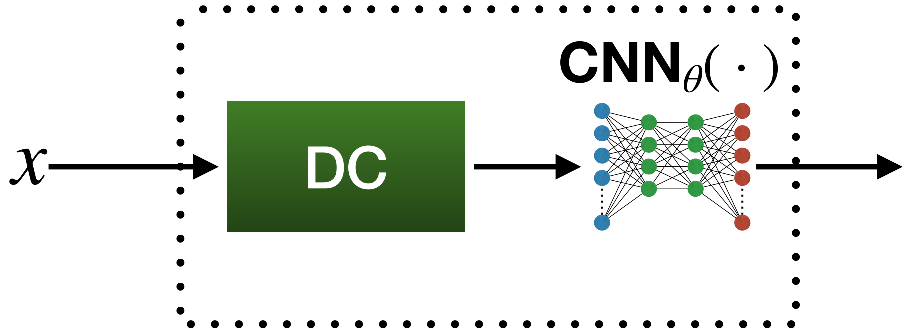
ex: plug-and-play/deep unrolling



$$f(x) = \textbf{CNN}(x - \eta A^\top(Ax - y))$$

Recent work on **"deep equilibrium models"** (Bai, Kolter, & Koltun 2019) has shown how to perform back-propagation on estimators implicitly defined through fixed-point equations.

Key Idea: Implicit Differentiation



$f_\theta(x) = \mathbf{CNN}_\theta(x - \eta A^\top(Ax - y))$

Fixed point: $x^* = f_\theta(x^*)$

Note: x^* implicitly a function of network parameters θ

$$\frac{\partial \ell}{\partial \theta}^\top = \frac{\partial \ell}{\partial x^*}^\top \frac{\partial x^*}{\partial \theta}$$

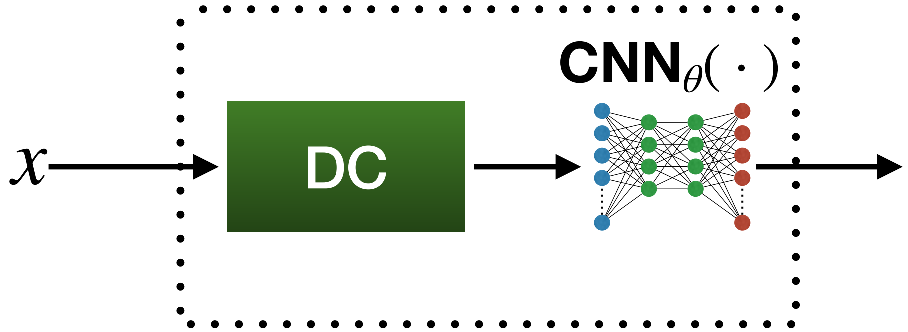
$$\frac{\partial x^*}{\partial \theta} = \frac{\partial f_\theta(x^*)}{\partial \theta} + \frac{\partial f_\theta(x^*)}{\partial x^*}\frac{\partial x^*}{\partial \theta}$$

$x^* \in \mathbb{R}^m$
$\theta \in \mathbb{R}^p$

$$\frac{\partial x^*}{\partial \theta} = \left(I - \frac{\partial f_\theta(x^*)}{\partial x^*}\right)^{-1} \frac{\partial f_\theta(x^*)}{d\theta}$$

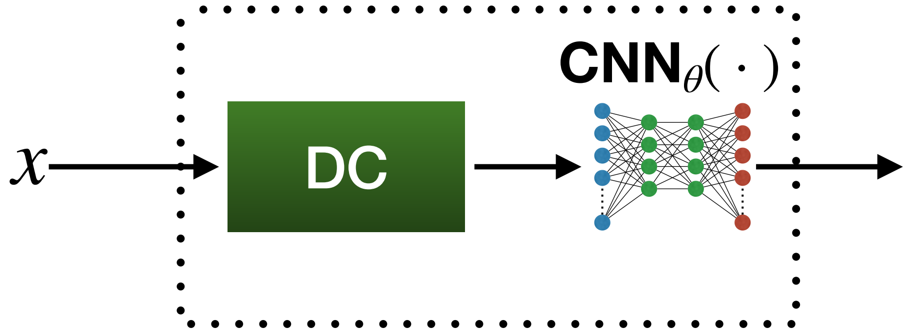
$m \times p$ $m \times m$ $m \times p$

Convergence to a fixed-point



$$f_\theta(x) = \mathbf{CNN}_\theta(x - \eta A^\top(Ax - y))$$

Convergence to a fixed-point

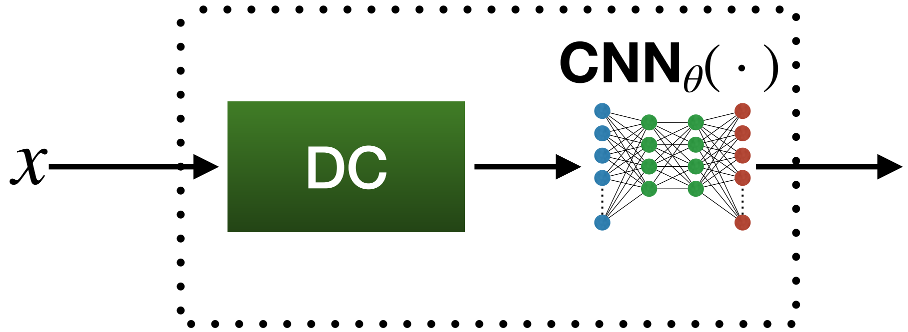


$$f_\theta(x) = \mathbf{CNN}_\theta(x - \eta A^\top(Ax - y))$$

- Q: Do the iterates converge to a fixed-point?

  A: If the map $f_\theta(\cdot)$ is **contractive**, yes, and convergence is linear.

Convergence to a fixed-point



$$f_\theta(x) = \mathbf{CNN}_\theta(x - \eta A^\top(Ax - y))$$

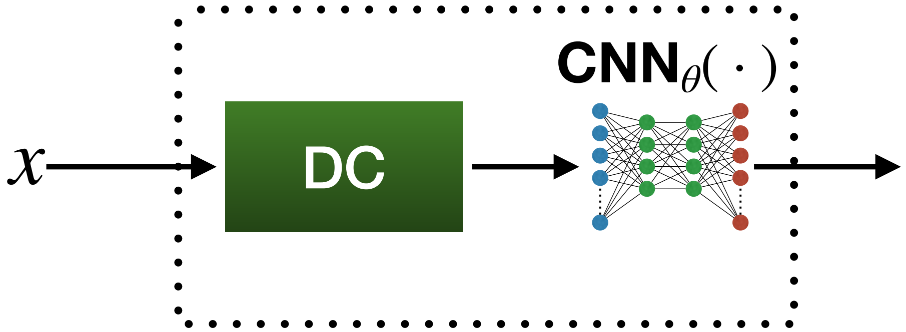
- Q: Do the iterates converge to a fixed-point?

  A: If the map $f_\theta(\cdot)$ is **contractive**, yes, and convergence is linear.

- Q: Can we guarantee contractivity?

  A: Yes! sufficient to bound **Lipschitz constant** of denoising CNN

Convergence to a fixed-point



$$f_\theta(x) = \mathbf{CNN}_\theta(x - \eta A^\top(Ax - y))$$

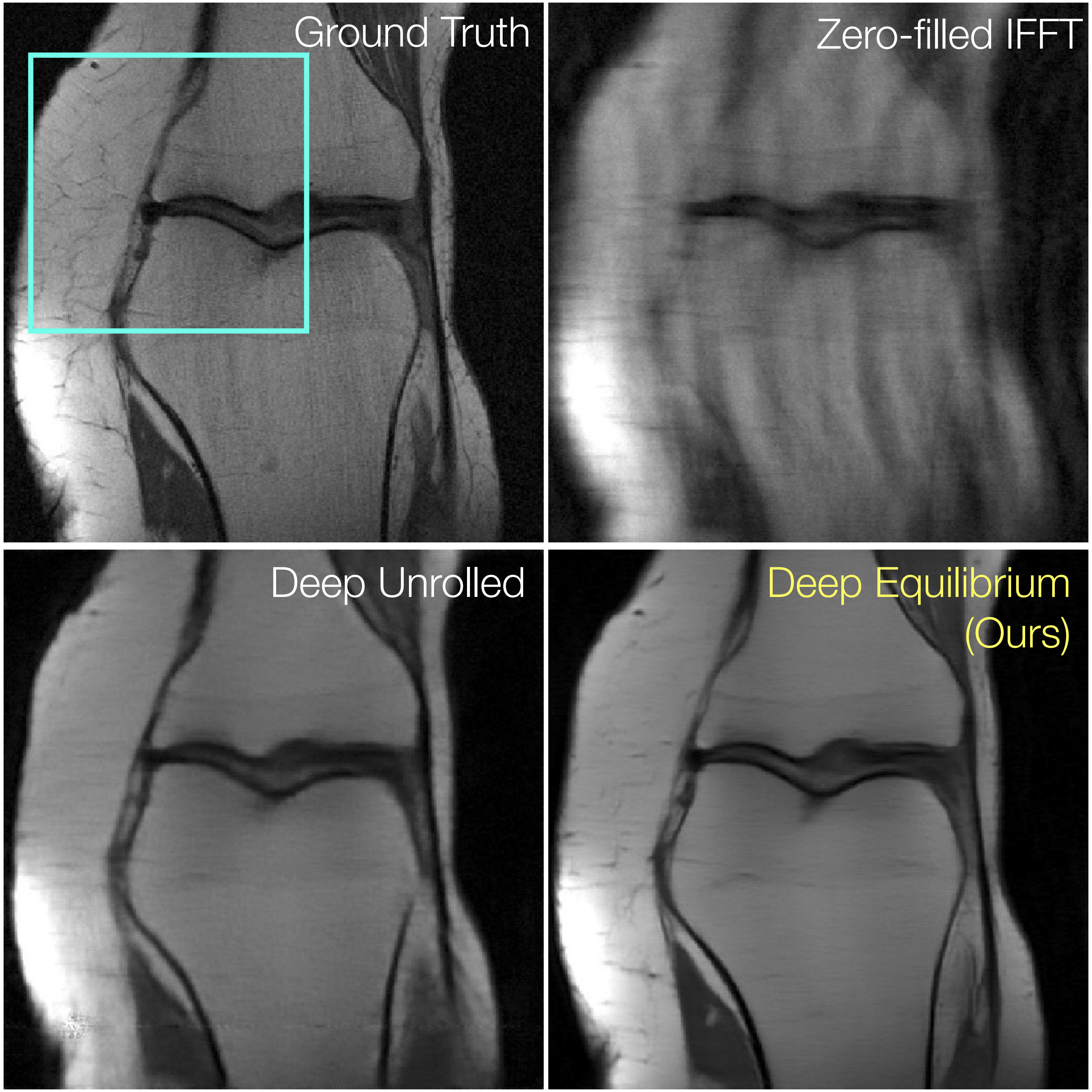
- Q: Do the iterates converge to a fixed-point?
  A: If the map $f_\theta(\cdot)$ is **contractive**, yes, and convergence is linear.
- Q: Can we guarantee contractivity?
  A: Yes! sufficient to bound **Lipschitz constant** of denoising CNN
- We use the "spectral normalization" technique of (Miyato et al., 2018)

MRI 8-fold Acceleration: Example Reconstruction

Data from FastMRI

Gilton, Ongie, & Willett 2021

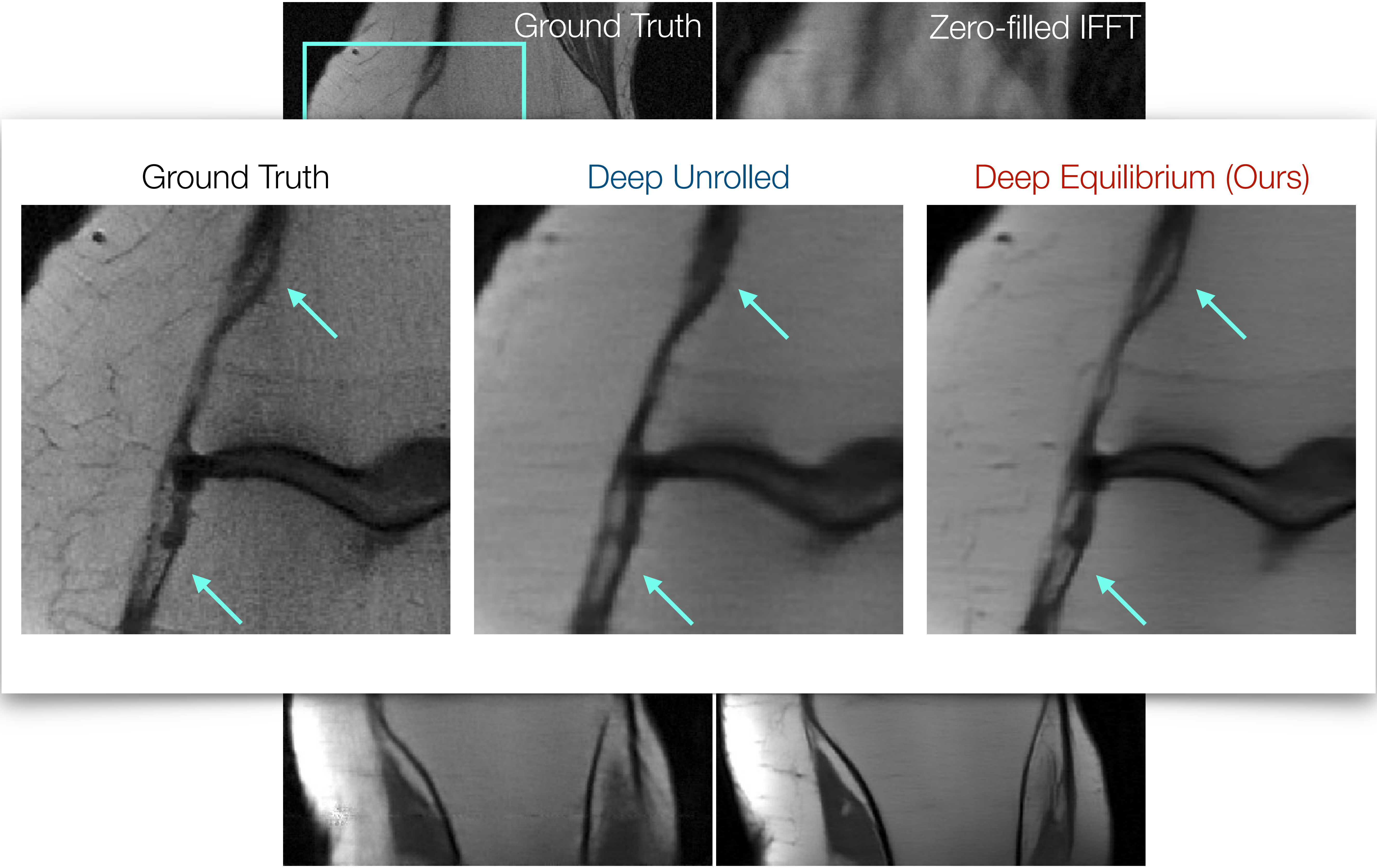
MRI 8-fold Acceleration: Example Reconstruction



Data from FastMRI

Gilton, Ongie, & Willett 2021

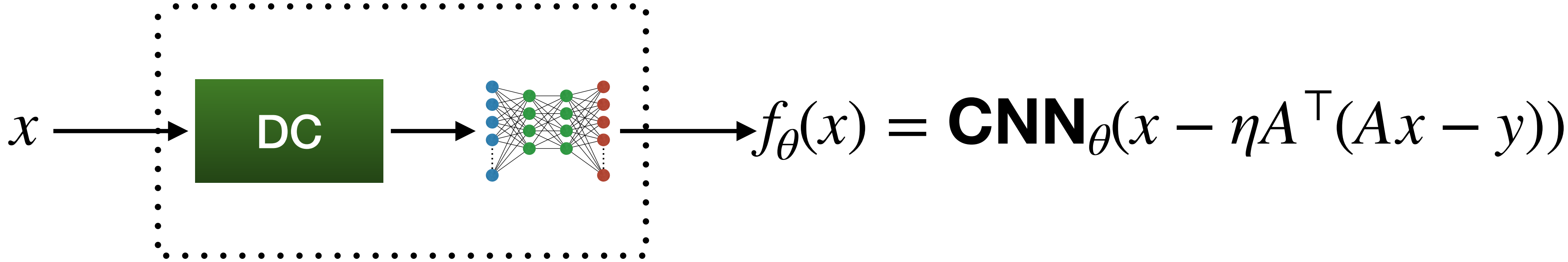
MRI 8-fold Acceleration: Example Reconstruction

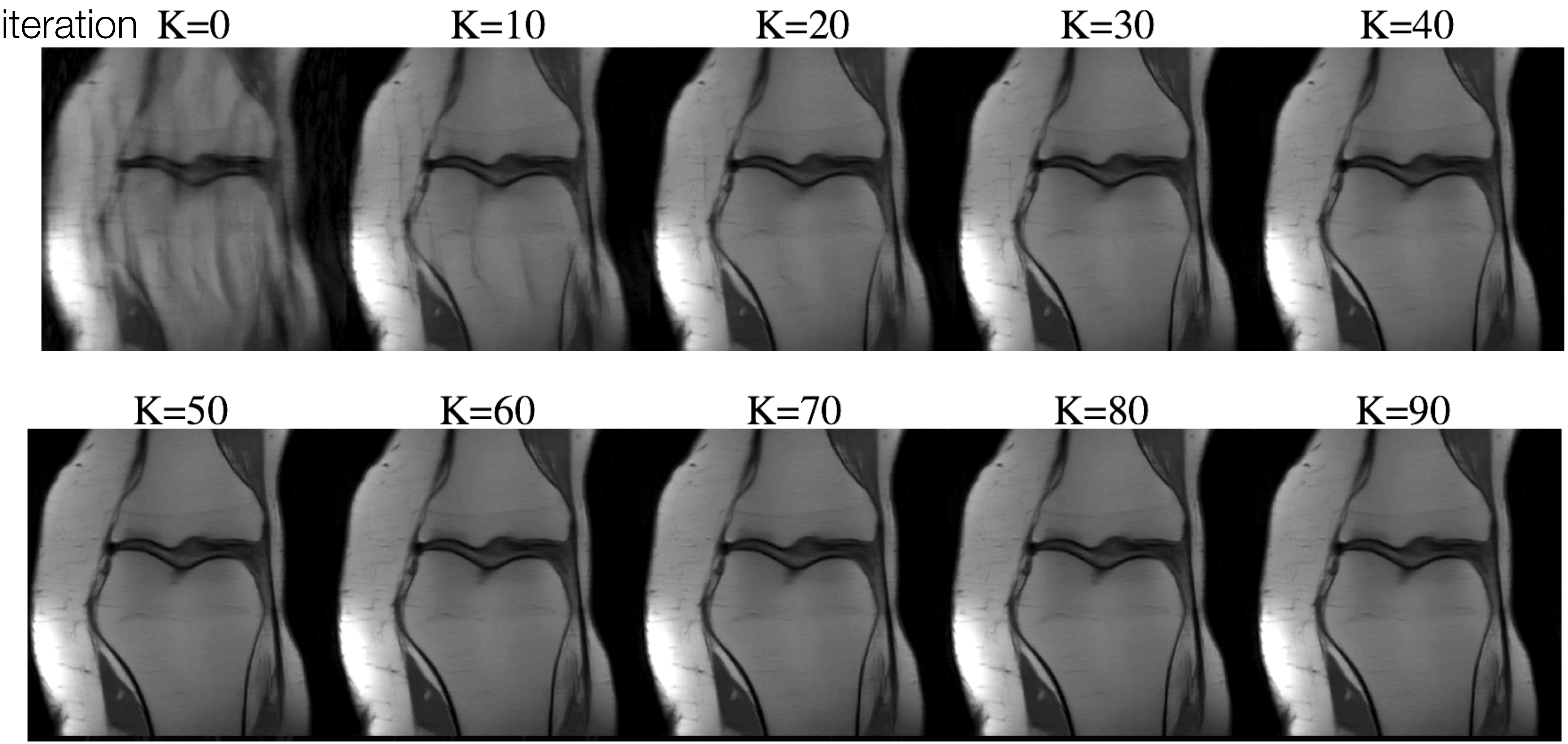


Data from FastMRI

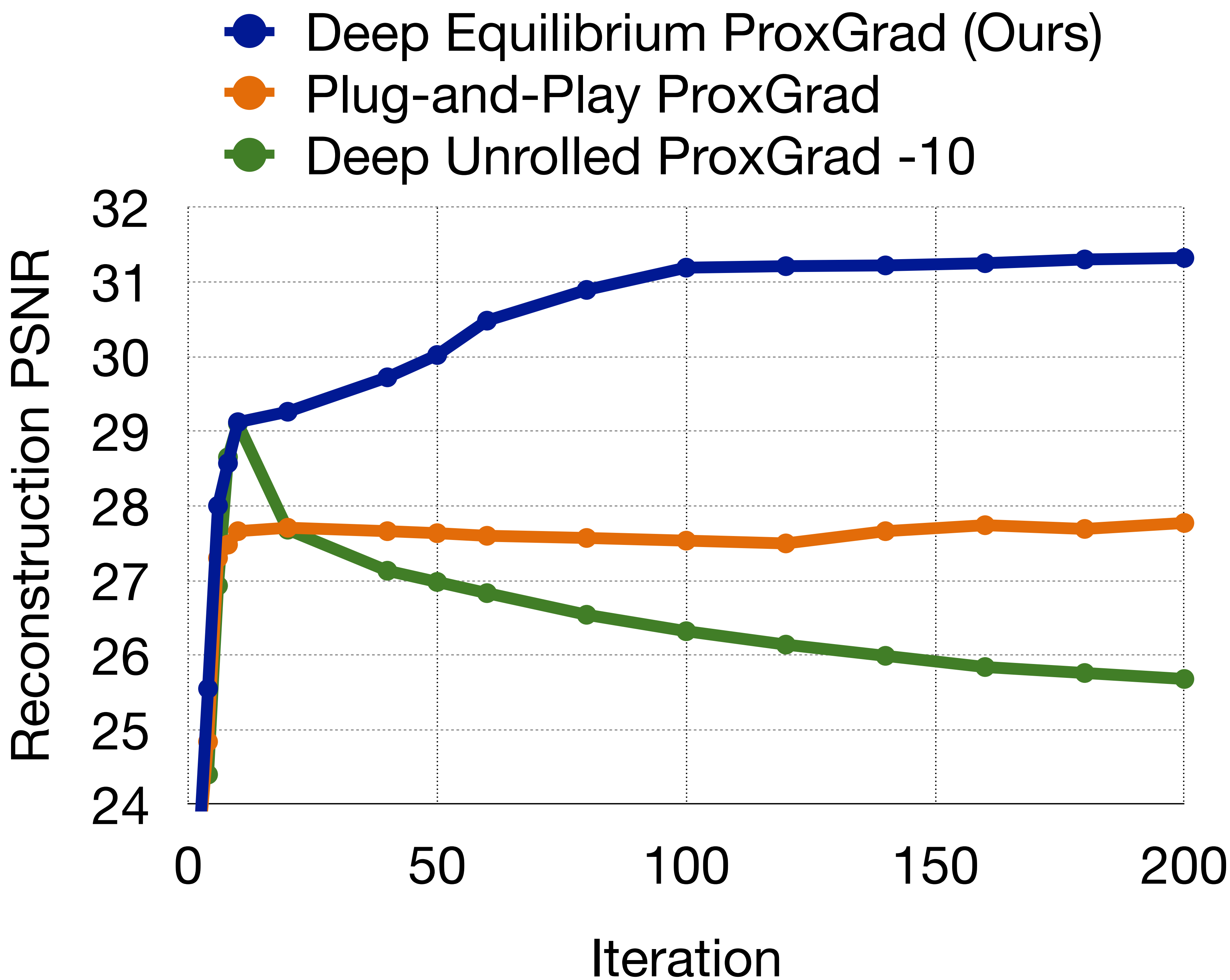
Gilton, Ongie, & Willett 2021

Deep Equilibrium — Illustration of Convergence

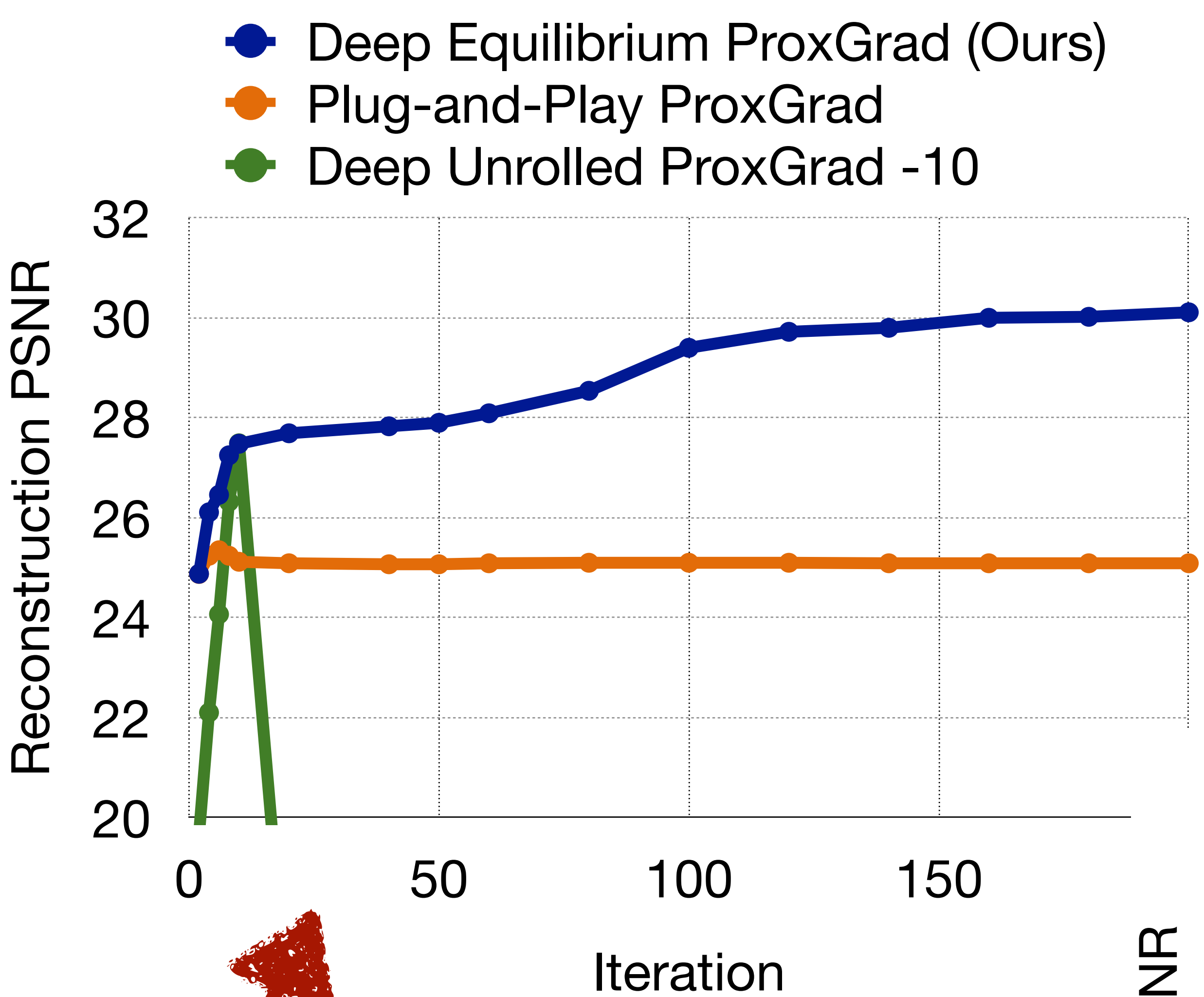




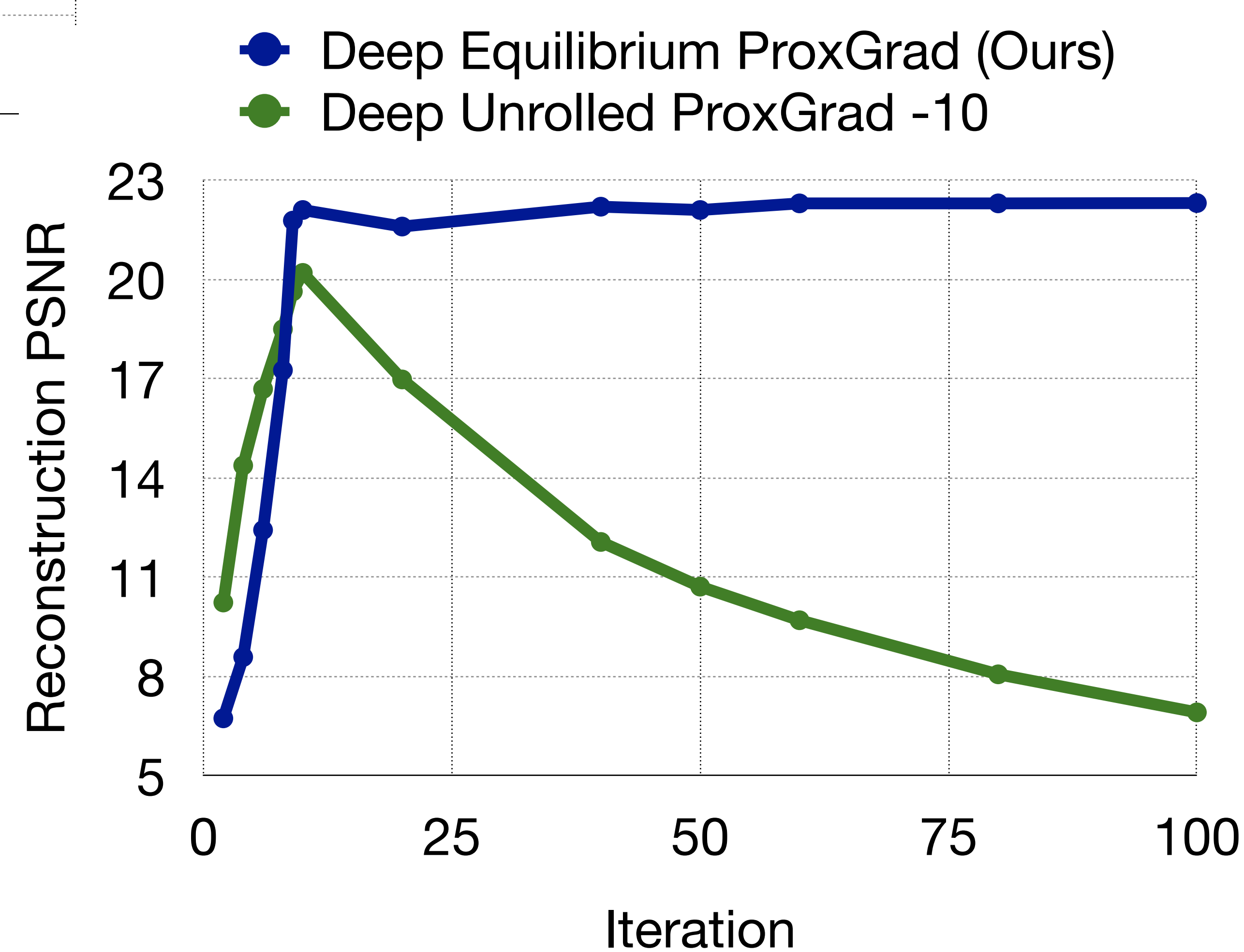
Deep Equilibrium — Illustration of Convergence (MRI)



Deep Equilibrium — Illustration of Convergence

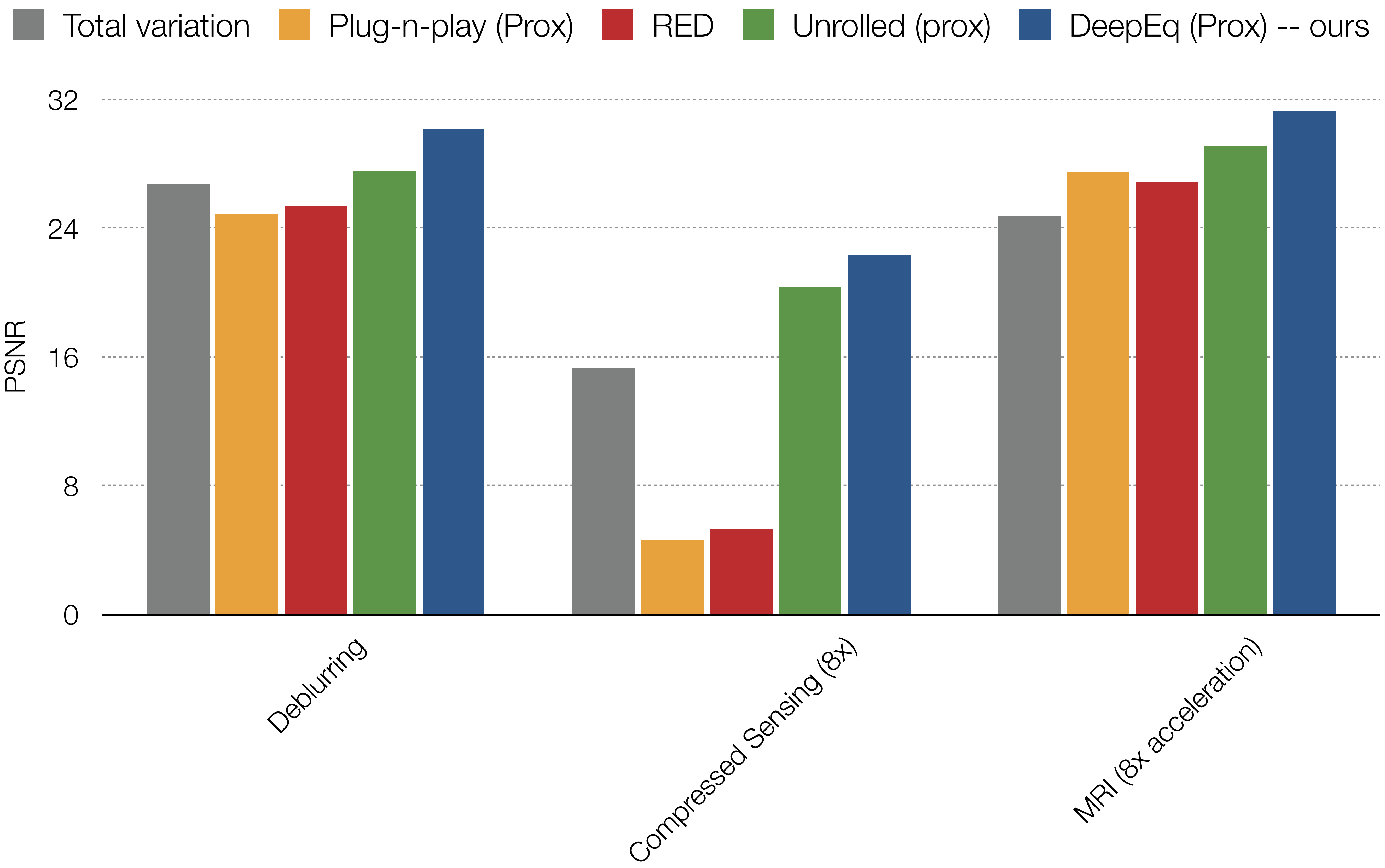


Compressed Sensing



Deblurring

Numerical results



Conclusion

Plug-and-play

Convergence Guarantees

Deep Unrolling

Reconstruction Accuracy

Deep Equilibrium

Pre-print on arXiv:

arXiv:2102.07944 [pdf, other] eess.IV cs.CV

Deep Equilibrium Architectures for Inverse Problems in Imaging

Authors: Davis Gilton, Gregory Ongie, Rebecca Willett

Paper also contains:

- Alternative Deep Equilibrium architectures based on
  - Gradient Descent
  - Alternating Directions Method of Multipliers (ADMM)

  with associated convergence guarantees.
- Anderson acceleration of fixed-point scheme
- Additional empirical results on MRI, deblurring, and compressed sensing

Key tradeoffs

- **Generality vs. sample complexity:**
  Leveraging known A during training gives lower sample complexity, but model must be retrained for each new A
- **Training stability vs. convergence guarantees:**
  Unrolled methods with a small number of blocks (K) are easier to train but lack convergence guarantees
- **Reconstruction accuracy vs. sensitivity to model mismatch:**
  Learning to solve for A_0 may yield a regularizer ill-suited to $A_1 \neq A_0$

Key tradeoffs

- **Generality vs. sample complexity:**
  Leveraging known A during training gives lower sample complexity, but model must be retrained for each new A
- **Training stability vs. convergence guarantees:**
  Unrolled methods with a small number of blocks (K) are easier to train but lack convergence guarantees
- **Reconstruction accuracy vs. sensitivity to model mismatch:**
  Learning to solve for A_0 may yield a regularizer ill-suited to $A_1 \neq A_0$

Methods trained for a specific forward model A outperform model-agnostic training when data is limited…

… but methods trained for a specific forward model A_0 break down when we transfer to a new forward model A_1

- MRI examples of model shift
  - Cartesian vs. non-Cartesian k-space sampling trajectories,
  - different undersampling factors,
  - different number of coils and coil sensitivity maps,
  - magnetic field inhomogeneity maps…

Image reconstruction by supervised learning — current paradigm

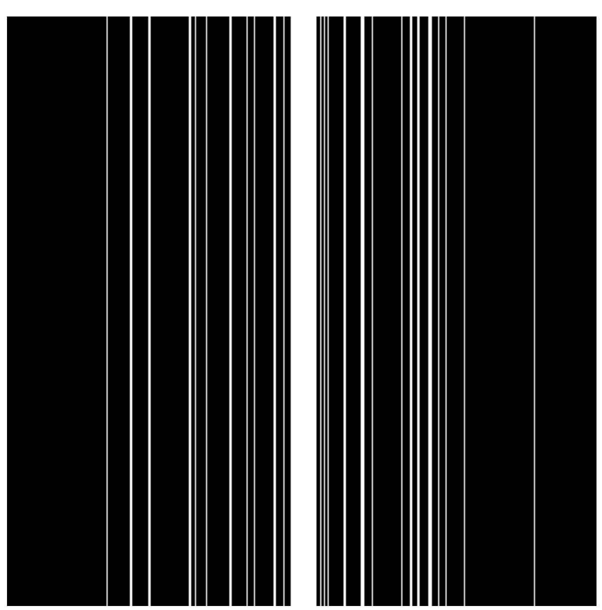
1. Collect and/or synthesize training data pairs (x_i, y_i) using a known forward model:

$$y_i = A_0 x_i + \varepsilon_i$$

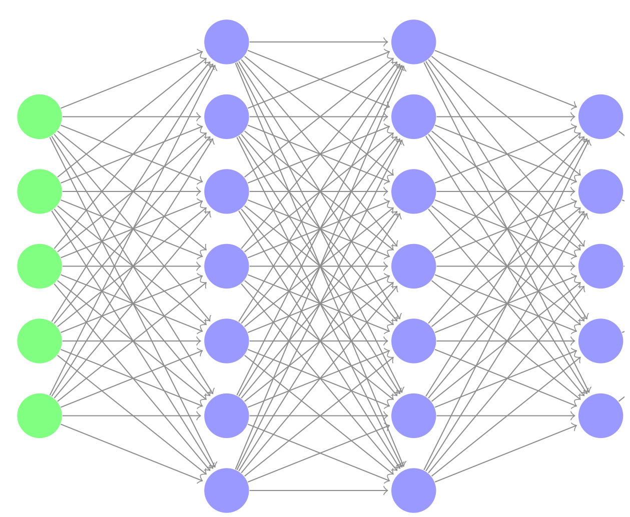
2. Train a reconstruction network f_θ by minimizing over a loss (e.g., MSE, SSIM)
3. Reconstruct new measurements y by $\hat{x} = f_\theta(y)$

e.g., accelerated MRI

k-space measurements y



reconstruction network $f_\theta(\cdot)$



reconstructed image $\hat{x} = f_\theta(y)$

Issue: Trained network is sensitive to changes in the forward model

Ground Truth

U-net Recon
same k-space
sampling
pattern as in training

PSNR: 31.7 dB

U-net Recon
different k-space sampling pattern than used in training

PSNR: 26.9 dB

~5 dB drop in PSNR

Effect originally observed in *Antun, Renna, Poon, Adcock, Hansen, 2019*

Potential solutions

- **Option 1: Retrain** the network from scratch using the new forward model.
  - Issue: Computationally costly
  - Issue: Might not have ground truth data to retrain (have y_i's but no $x_i's$)

- **Option 2:** Train on an **ensemble** of forward models.
  - Issue: High-dimensional set to sample from (e.g., all possible k-space masks)
  - Issue: Numerical evidence suggests this gives *worse* performance overall

- **Option 3:** Use a **model agnostic** approach (e.g., Plug-and-Play or GAN)
  - Issue: May not have enough ground truth images to learn a sufficiently expressive model

- **Option 4 (Proposed): Adapt** the network to solve the new inverse problem

Model Adaptation for Image Reconstruction

Given a reconstruction network f_θ trained to solve an inverse problem

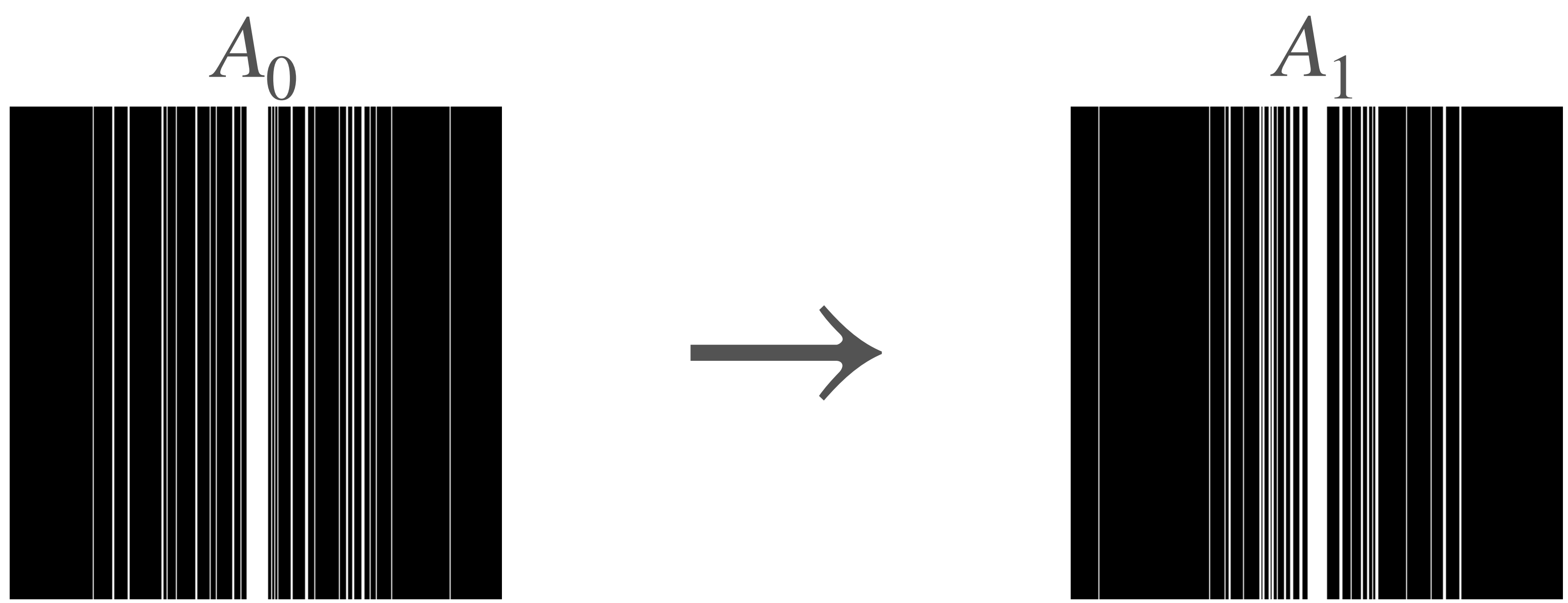
$$y = A_0 x + \varepsilon$$

adapt/retrain/augment it to solve a new inverse problem

$$y = A_1 x + \varepsilon$$

(In this talk, assume new forward model A_1 is known.)

E.g.



train on this
k-space sampling mask

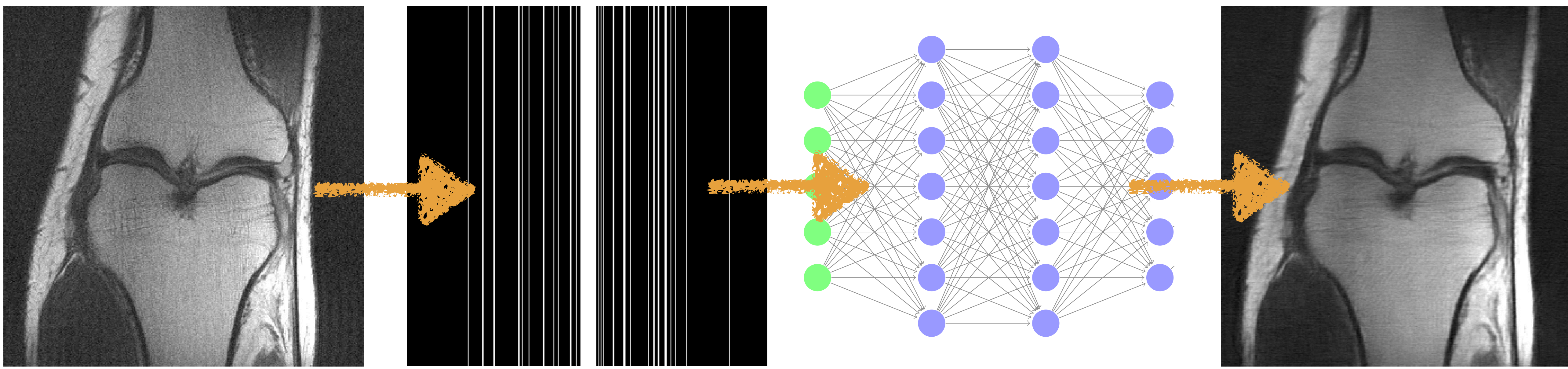
deploy on this
k-space sampling mask

Model adaptation

Key Idea:

The composition $f_\theta \circ A_0$ acts as an auto-encoder, $f_\theta(A_0 x) \approx x$

original image	simulated measurements	reconstruction network	reconstructed image
x	$A_0 x$	$f_\theta(\cdot)$	$f_\theta(A_0 x)$



Use auto-encoder property as a **prior/regularizer** in an iterative model-based reconstruction scheme

Model adaptation without calibration data

- We adopt a **regularization-by-denoising (RED)** approach using proximal gradient descent as base algorithm and $f_\theta \circ A_0$ as our "denoiser".

- Motivated by cost function:

$$\min_x \|A_1 x - y\|_2^2 + \lambda\, x^\top (x - f_\theta(A_0 x))$$

regularization parameter λ allows us trade-off between data-consistency and regularization

- Proposed Iterative algorithm:

$$z^{(k)} = (A_1^\top A_1 + \lambda I)^{-1}(A_1^\top y + \lambda x^{(k)})$$

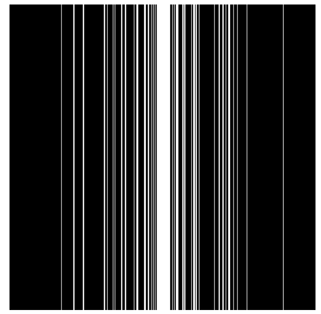
data consistency step

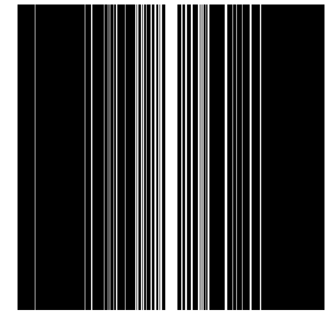
$$x^{(k+1)} = f_\theta(A_0 z^{(k)})$$

reuse the pre-trained network to regularize

RED (Romano, Elad, & Milanfar 2017)

Illustration on FastMRI knee data: 6x → 6x acceleration

A_0 : 

A_1 : 

different sampling pattern
same number of lines

Ground Truth

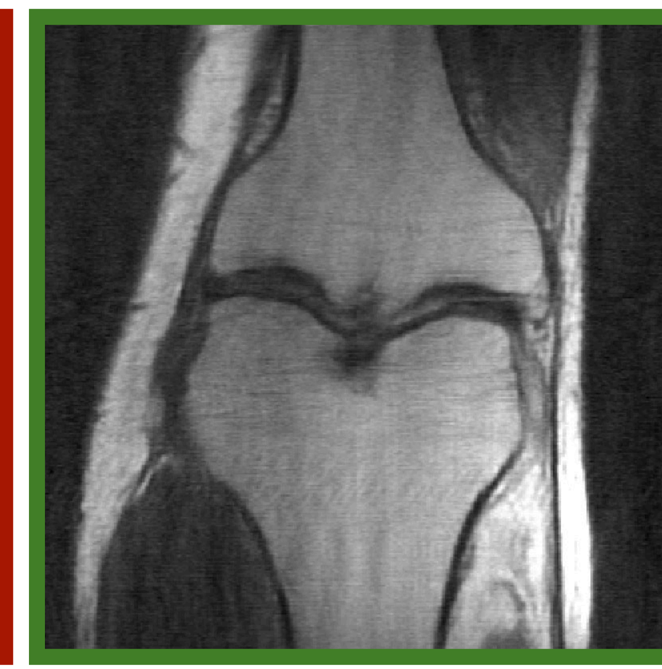
U-net Recon
Train on A_0/Test on A_0

PSNR: 31.7 dB
SSIM: 0.80

No Model Adaptation
Train on A_0/Test on A_1

PSNR: 26.9 dB
SSIM: 0.76

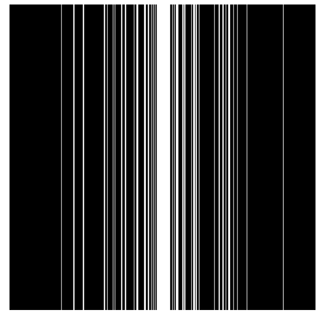
Model Adaptation
Train on A_0/Test on A_1



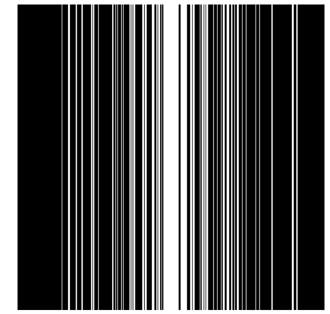
PSNR: 31.3 dB
SSIM: 0.78

Illustration on FastMRI knee data: 6x → **4x** acceleration

A_0 :



A_1 :



sample more lines

Ground Truth

U-net Recon

Train on A_0/Test on A_0

PSNR: 31.7 dB
SSIM: 0.80

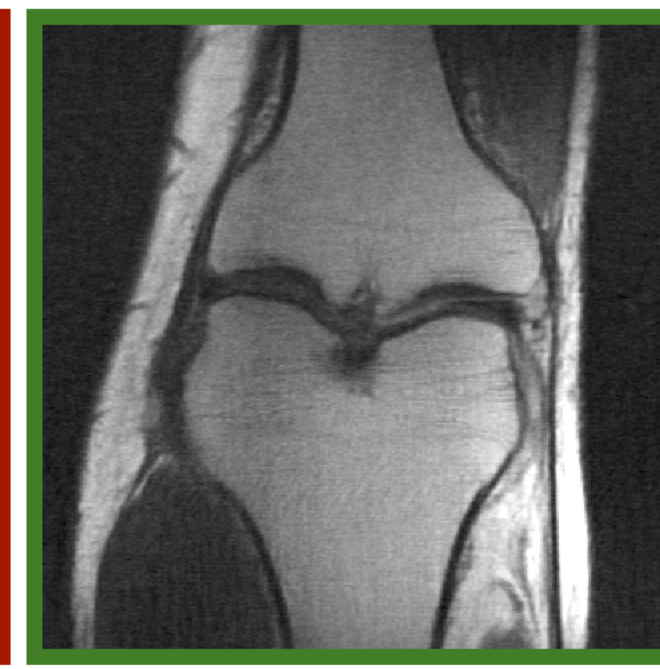
No Model Adaptation

Train on A_0/Test on A_1

PSNR: 26.4 dB
SSIM: 0.80

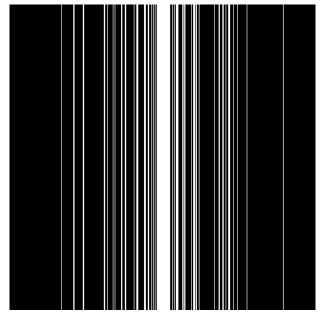
Model Adaptation

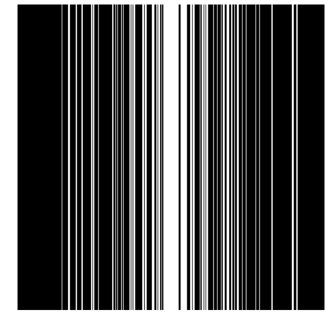
Train on A_0/Test on A_1



PSNR: 33.0 dB
SSIM: 0.83

Illustration on FastMRI knee data: 6x → **4x** acceleration

A_0 : 

A_1 : 

sample more lines

Ground Truth

U-net Recon

Train on A_1/Test on A_1

PSNR: 33.5 dB

SSIM: 0.83

No Model Adaptation

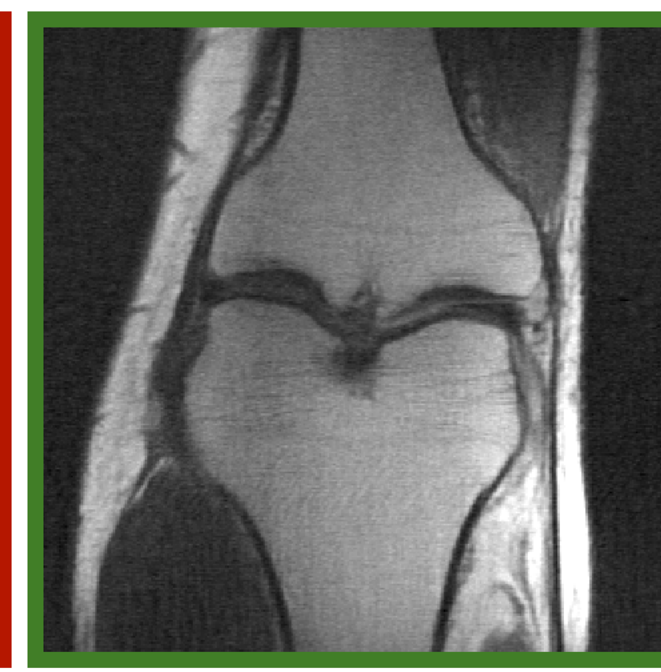
Train on A_0/Test on A_1

PSNR: 26.4 dB

SSIM: 0.80

Model Adaptation

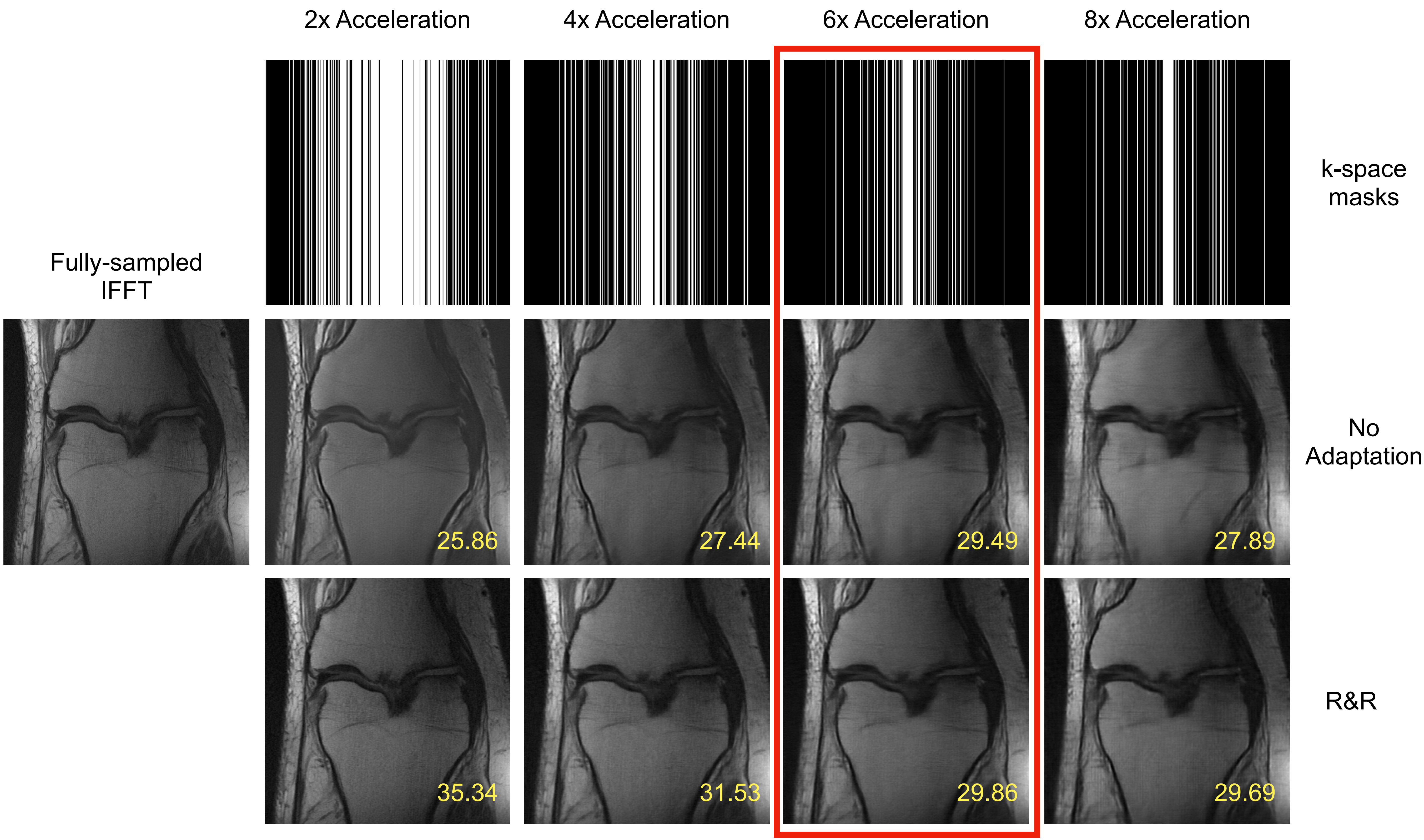
Train on A_0/Test on A_1



PSNR: 33.0 dB

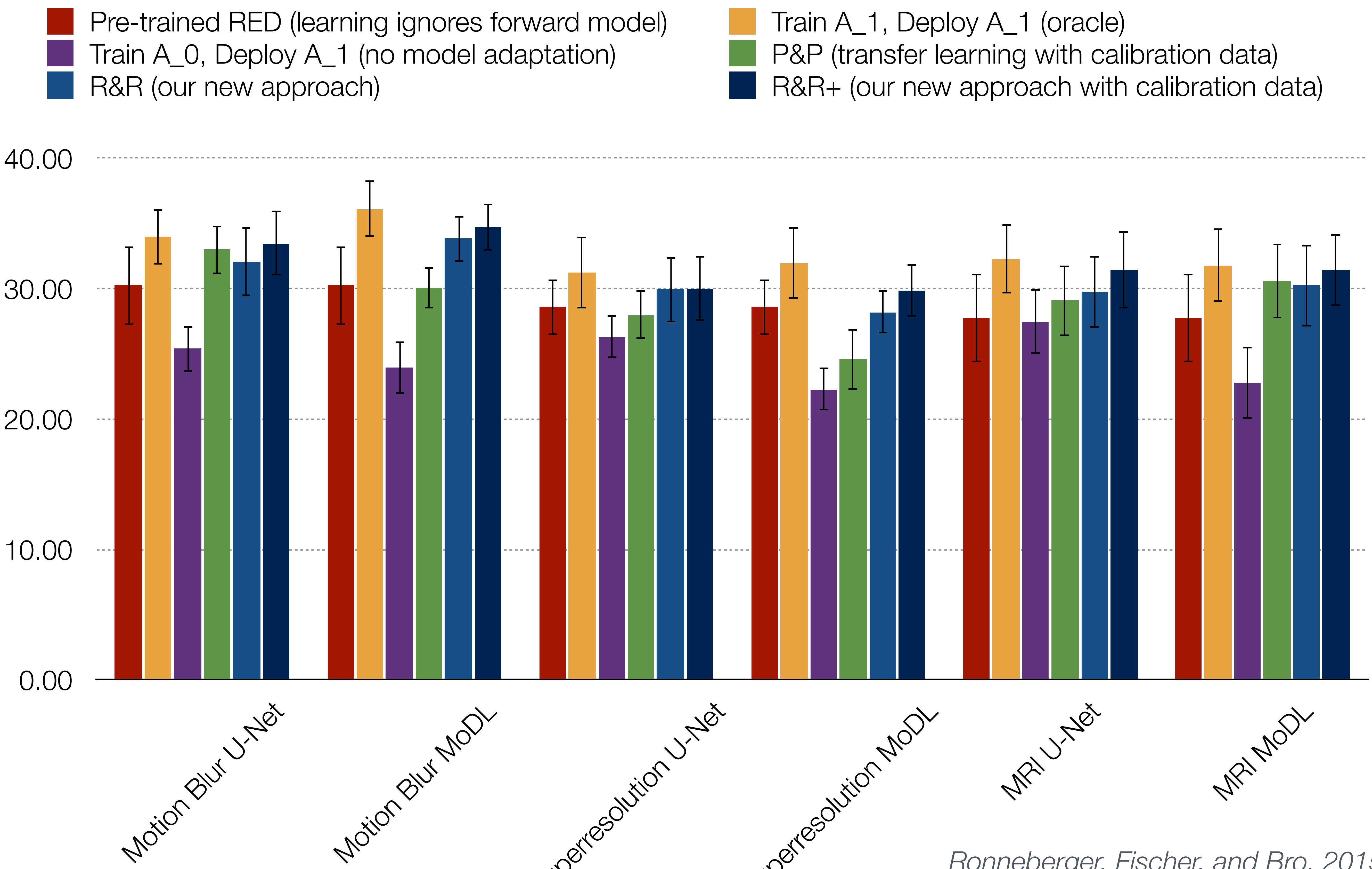
SSIM: 0.83

MRI Example



2x Acceleration
4x Acceleration
6x Acceleration
8x Acceleration
k-space masks
Fully-sampled IFFT
No Adaptation
R&R
25.86
27.44
29.49
27.89
35.34
31.53
29.86
29.69

Performance with known A_1



Ronneberger, Fischer, and Bro, 2015
Jure Zbontar, Florian Knoll, and others 2019

Summary

- We introduce the problem of **model adaptation for learned image reconstruction**
- Propose two solutions:
  - With calibration data: transfer learning approach (P&P)
  - Without calibration data: turn pre-trained network into regularizer (R&R)
- arXiv:2012.00139 [pdf, other] eess.IV cs.CV

  Model Adaptation for Inverse Problems in Imaging

  Authors: Davis Gilton, Gregory Ongie, Rebecca Willett

    - Extensions to the case where the new forward model A_1 is unknown
    - An alternative approach for model adaptation with calibration data (R&R+)
    - Comparisons with other baselines (e.g., TV, RED with other denoisers, GAN's)
    - Results on other inverse problems, including deblurring and super-resolution.

Thank you!