Deep learning and Inverse Problems

Daniel Otero Baguer, Peter Maass

Center for Industrial Mathematics (ZeTeM)
University of Bremen

Marseille
14-15.11.2019

Outline

Section 1

Introduction

Inverse Problems

Consider an operator $A : X \to Y$ between Hilbert spaces X and Y.
Given measured noisy data

$$y^\delta = Ax^\dagger + \tau, \tag{1}$$

where τ, with $\|\tau\| \leq \delta$, describes the noise in the measurement

Aim: Obtain an approximation $\hat{x}$ for $x^\dagger$

Inverse Problems

Consider an operator $A : X \to Y$ between Hilbert spaces X and Y. Given measured noisy data

$$y^\delta = Ax^\dagger + \tau, \tag{1}$$

where τ, with $\|\tau\| \leq \delta$, describes the noise in the measurement

Aim: Obtain an approximation $\hat{x}$ for $x^\dagger$

Example: Computed Tomography

Radon transform

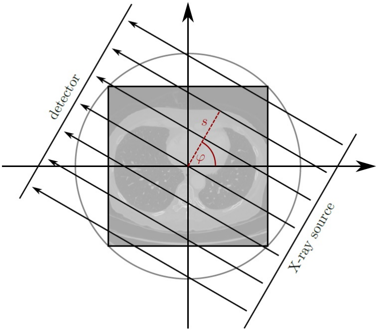


Figure: Parallel beam geometry

Example: Computed Tomography

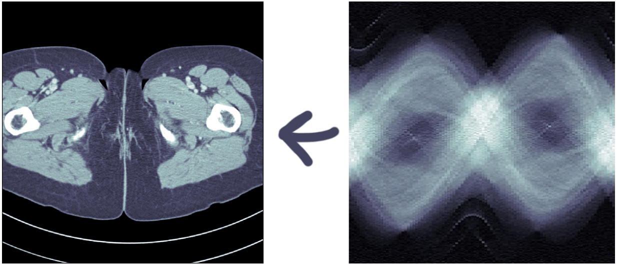


Figure: Human phantom and corresponding sinogram

Classical approaches

- TSVD
- Tikhonov
- Landweber
- **Variational regularization**:

$$T_\alpha(y^\delta) = \arg\min \frac{1}{2}\|Ax - y^\delta\|^2 + \alpha R(x) \qquad (2)$$

Examples of hand-crafted regularizers: $\|x\|^2$, $\|x\|_1$, $\|\nabla x\|_1$

How to choose R and the regularization parameters, e.g α?

Classical approaches

- TSVD
- Tikhonov
- Landweber
- **Variational regularization**:

$$T_\alpha(y^\delta) = \arg\min \frac{1}{2}\|Ax - y^\delta\|^2 + \alpha R(x) \tag{2}$$

Examples of hand-crafted regularizers: $\|x\|^2$, $\|x\|_1$, $\|\nabla x\|_1$

How to choose R and the regularization parameters, e.g α?

Data-driven approaches[1]

Assume training data is given: $\{x_i^\dagger,\ y_i^\delta\}_{i=1}^N$

Data-driven parameter choice:

$$\hat{\alpha} = \operatorname*{arg\,min}_{\alpha \in \mathbb{R}_+} \sum_{i=1}^{N} \ell(T_\alpha(y_i^\delta),\ x_i^\dagger) \tag{3}$$

Data-driven regularized inverse $T_\Theta : Y \to X$

[1] Simon Arridge, Peter Maass, Ozan Öktem, and Carola-Bibiane Schönlieb. "Solving inverse problems using data-driven models". In: *Acta Numerica* 28 (2019), pp. 1–174.

Data-driven approaches[1]

Assume training data is given: $\{x_i^\dagger,\ y_i^\delta\}_{i=1}^N$

Data-driven parameter choice:

$$\hat{\alpha} = \operatorname*{arg\,min}_{\alpha \in \mathbb{R}_+} \sum_{i=1}^{N} \ell(T_\alpha(y_i^\delta),\ x_i^\dagger) \tag{3}$$

Data-driven regularized inverse $T_\Theta : Y \to X$

[1] Simon Arridge, Peter Maass, Ozan Öktem, and Carola-Bibiane Schönlieb. "Solving inverse problems using data-driven models". In: *Acta Numerica* 28 (2019), pp. 1–174.

Section 2

Data-driven approaches

Recent approaches

- Learned methods

$$T_\Theta : Y \to X \tag{4}$$

- Learned regularizers

$$T_\Theta(y^\delta) = \arg\min_{x \in X} \ell(Ax,\ y^\delta) + R_\Theta(x) \tag{5}$$

- Generative Networks

$$T_\Theta(y^\delta) = \arg\min_{z \in Z} \ell(A\varphi_\Theta(z),\ y^\delta) + R_\Theta(z) \tag{6}$$

Recent approaches

- **Learned methods**

$$T_\Theta : Y \to X \tag{7}$$

- Learned regularizers

$$T_\Theta(y^\delta) = \arg\min_{x \in X} \ell(Ax,\ y^\delta) + R_\Theta(x) \tag{8}$$

- Generative Networks

$$T_\Theta(y^\delta) = \arg\min_{z \in Z} \ell(A\varphi_\Theta(z),\ y^\delta) + R_\Theta(z) \tag{9}$$

Learned methods

- [2] Fully learned
- [3] Learned post-processing: $T_\Theta = \mathcal{F}_\Theta \circ A^\dagger$
- [4] Learned iterative schemes

Training: Takes quite some time (even weeks)
Evaluation: Takes milliseconds

[2] Bo Zhu, Jeremiah Z. Liu, Stephen F. Cauley, Bruce R. Rosen, and Matthew S. Rosen. "Image reconstruction by domain-transform manifold learning". In: *Nature* (2018). URL: https://doi.org/10.1038/nature25988.

[3] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. "Deep Convolutional Neural Network for Inverse Problems in Imaging". In: *IEEE Transactions on Image Processing* 26.9 (Sept. 2017), pp. 4509–4522. ISSN: 1057-7149. DOI: 10.1109/TIP.2017.2713099.

[4] Andreas Hauptmann, Felix Lucka, Marta Betcke, Nam Huynh, Jonas Adler, Ben Cox, Paul Beard, Sebastien Ourselin, and Simon Arridge. "Model-Based Learning for Accelerated, Limited-View 3-D Photoacoustic Tomography". In: *IEEE transactions on medical imaging* 37.6 (2018), pp. 1382–1393.

Learned methods

- [2] Fully learned
- [3] Learned post-processing: $T_\Theta = \mathcal{F}_\Theta \circ A^\dagger$
- [4] Learned iterative schemes

Training: Takes quite some time (even weeks)
Evaluation: Takes milliseconds

[2] Bo Zhu, Jeremiah Z. Liu, Stephen F. Cauley, Bruce R. Rosen, and Matthew S. Rosen. "Image reconstruction by domain-transform manifold learning". In: *Nature* (2018). URL: https://doi.org/10.1038/nature25988.

[3] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. "Deep Convolutional Neural Network for Inverse Problems in Imaging". In: *IEEE Transactions on Image Processing* 26.9 (Sept. 2017), pp. 4509–4522. ISSN: 1057-7149. DOI: 10.1109/TIP.2017.2713099.

[4] Andreas Hauptmann, Felix Lucka, Marta Betcke, Nam Huynh, Jonas Adler, Ben Cox, Paul Beard, Sebastien Ourselin, and Simon Arridge. "Model-Based Learning for Accelerated, Limited-View 3-D Photoacoustic Tomography". In: *IEEE transactions on medical imaging* 37.6 (2018), pp. 1382–1393.

Learned methods: Fully learned

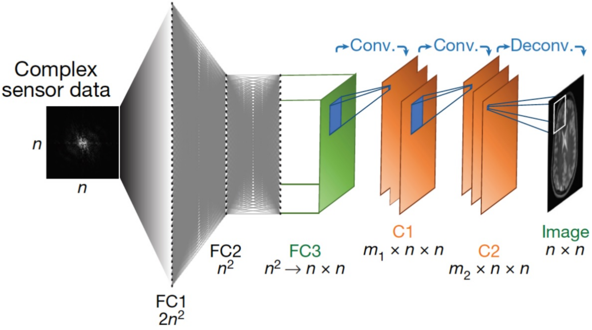


Figure: The AUTOMAP[5] architecture

[5] Bo Zhu, Jeremiah Z. Liu, Stephen F. Cauley, Bruce R. Rosen, and Matthew S. Rosen. "Image reconstruction by domain-transform manifold learning". In: *Nature* (2018). URL: https://doi.org/10.1038/nature25988.

Learned methods: Post-processing

Given data pairs $\{(y_i^\delta, x_i^\dagger)\}$ and a pseudo inverse $A^\dagger$:

- Train a network $\mathcal{F}_\Theta : X \to X$ by minimizing

$$\min_\theta \frac{1}{N} \sum_{i=1}^{N} \|\mathcal{F}_\Theta(A^\dagger y_i^\delta) - x_i^\dagger\|^2 \tag{10}$$

Remark: Is similar to denoising ($A^\dagger y_i^\delta$ noisy version of $x^\dagger$)

Architecture: Autoencoder-like

Result: $T_\Theta(y^\delta) = \mathcal{F}_\Theta(A^\dagger y^\delta)$

Learned methods: Post-processing

Given data pairs $\{(y_i^\delta, x_i^\dagger)\}$ and a pseudo inverse $A^\dagger$:

- Train a network $\mathcal{F}_\Theta : X \to X$ by minimizing

$$\min_\theta \frac{1}{N} \sum_{i=1}^{N} \|\mathcal{F}_\Theta(A^\dagger y_i^\delta) - x_i^\dagger\|^2 \tag{10}$$

Remark: Is similar to denoising ($A^\dagger y_i^\delta$ noisy version of $x^\dagger$)

Architecture: Autoencoder-like

Result: $T_\Theta(y^\delta) = \mathcal{F}_\Theta(A^\dagger y^\delta)$

Learned methods: Post-processing

Given data pairs $\{(y_i^\delta, x_i^\dagger)\}$ and a pseudo inverse $A^\dagger$:

- Train a network $\mathcal{F}_\Theta : X \to X$ by minimizing

$$\min_\theta \frac{1}{N} \sum_{i=1}^{N} \|\mathcal{F}_\Theta(A^\dagger y_i^\delta) - x_i^\dagger\|^2 \tag{10}$$

Remark: Is similar to denoising ($A^\dagger y_i^\delta$ noisy version of $x^\dagger$)

Architecture: Autoencoder-like

Result: $T_\Theta(y^\delta) = \mathcal{F}_\Theta(A^\dagger y^\delta)$

Learned methods: Post-processing

Given data pairs $\{(y_i^\delta, x_i^\dagger)\}$ and a pseudo inverse $A^\dagger$:

- Train a network $\mathcal{F}_\Theta : X \to X$ by minimizing

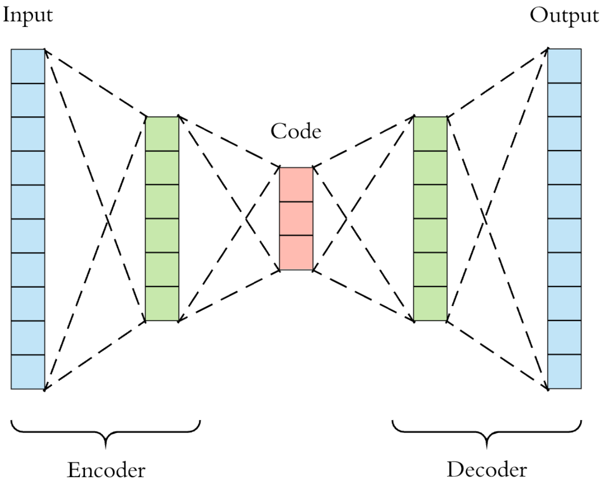
$$\min_\theta \frac{1}{N} \sum_{i=1}^{N} \|\mathcal{F}_\Theta(A^\dagger y_i^\delta) - x_i^\dagger\|^2 \tag{10}$$

Remark: Is similar to denoising ($A^\dagger y_i^\delta$ noisy version of $x^\dagger$)

Architecture: Autoencoder-like

Result: $T_\Theta(y^\delta) = \mathcal{F}_\Theta(A^\dagger y^\delta)$

Autoencoder



U-Net

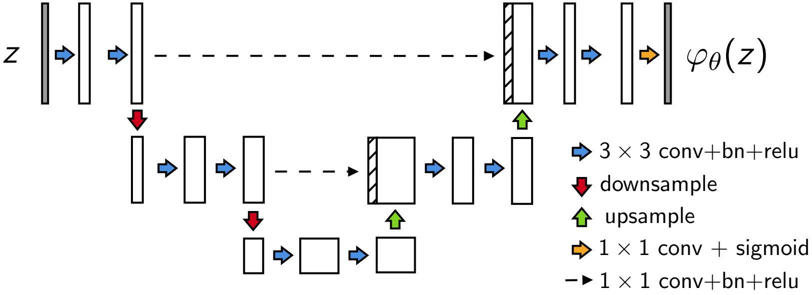
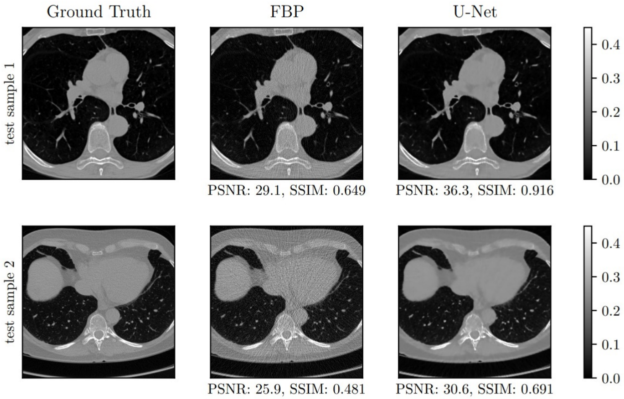


Figure: U-Net architecture

Post-processing for CT[6]



[6] Johannes Leuschner, Maximilian Schmidt, Daniel Otero Baguer, and Peter Maass. *The LoDoPaB-CT Dataset: A Benchmark Dataset for Low-Dose CT Reconstruction Methods*. 2019. arXiv: 1910.01113 [eess.IV].

The LoDoPaB-CT Dataset and DIV$\alpha\ell$

The LoDoPaB-CT Dataset

A Benchmark Dataset for Low-Dose CT Reconstruction Methods

Johannes Leuschner *[1], Maximilian Schmidt †[1], Daniel Otero Baguer ‡[1], and Peter Maaß §[1]

[1]University of Bremen, Center for Industrial Mathematics

October 2019

Abstract

Deep Learning approaches for solving Inverse Problems in imaging have become very effective and are demonstrated to be quite competitive in the field. Comparing these approaches is a challenging task since they highly rely on the data and the setup that is used for training. We

Learned methods: Learned iterative schemes

Inspired from iterative optimization methods

$$\hat{x} = \arg\min \frac{1}{2}\|Ax - y^\delta\| + \alpha R(x) \tag{11}$$

Proximal gradient algorithm:

$$x^{k+1} = \underset{R,\alpha,\lambda}{\mathrm{Prox}}(x^k - \lambda A^*(Ax^k - y^\delta)) \tag{12}$$

More general:

$$x^{k+1} = \varphi_\Theta(x^k,\ A^*(Ax^k - y^\delta)) \tag{13}$$

Learned methods: Learned iterative schemes

Take a small fixed number of iterations $k = 1, 2, \cdots, L$, e.g. $L = 10$

Then let $T_\Theta : Y \to X$ be

$$T_\Theta(y^\delta) = x^L \tag{14}$$

with

$$x^0 \leftarrow \text{initialized random} \tag{15}$$

$$x^{k+1} = \varphi_\Theta(x^k,\ A^*(Ax^k - y^\delta)) \tag{16}$$

Optimize Θ:

$$\hat{\Theta} = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \| T_\Theta(y_i^\delta) - x_i^\dagger \|^2 \tag{17}$$

Learned methods: Learned iterative schemes

Take a small fixed number of iterations $k = 1, 2, \cdots, L$, e.g. $L = 10$

Then let $T_\Theta : Y \to X$ be

$$T_\Theta(y^\delta) = x^L \tag{14}$$

with

$$x^0 \leftarrow \text{initialized random} \tag{15}$$

$$x^{k+1} = \varphi_\Theta(x^k,\ A^*(Ax^k - y^\delta)) \tag{16}$$

Optimize Θ:

$$\hat{\Theta} = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \| T_\Theta(y_i^\delta) - x_i^\dagger \|^2 \tag{17}$$

Recent approaches

- Learned methods

$$T_\Theta : Y \to X \tag{18}$$

- **Learned regularizers**

$$T_\Theta(y^\delta) = \arg\min_{x \in X} \ell(Ax,\ y^\delta) + R_\Theta(x) \tag{19}$$

- Generative Networks

$$T_\Theta(y^\delta) = \arg\min_{z \in Z} \ell(A\varphi_\Theta(z),\ y^\delta) + R_\Theta(z) \tag{20}$$

Learned regularizers

- [7] NETT (Network Tikhonov)
- [8] Adversarial regularizer

Training: Takes quite some time (even weeks)
Evaluation: Takes minutes
Data: Unsupervised data

[7] Housen Li, Johannes Schwab, Stephan Antholzer, and Markus Haltmeier. "NETT: Solving Inverse Problems with Deep Neural Networks". In: *arXiv preprint arXiv:1803.00092* (Feb. 2018).

[8] Sebastian Lunz, Ozan Öktem, and Carola-Bibiane Schönlieb. "Adversarial Regularizers in Inverse Problems". In: *arXiv preprint arXiv:1805.11572* (2018).

Learned regularizers

- [7] NETT (Network Tikhonov)
- [8] Adversarial regularizer

Training: Takes quite some time (even weeks)
Evaluation: Takes minutes
Data: Unsupervised data

[7] Housen Li, Johannes Schwab, Stephan Antholzer, and Markus Haltmeier. "NETT: Solving Inverse Problems with Deep Neural Networks". In: *arXiv preprint arXiv:1803.00092* (Feb. 2018).

[8] Sebastian Lunz, Ozan Öktem, and Carola-Bibiane Schönlieb. "Adversarial Regularizers in Inverse Problems". In: *arXiv preprint arXiv:1805.11572* (2018).

Learned regularizers: NETT

NETT (Network Tikhonov)

$$T_\Theta(y^\delta) = \operatorname*{arg\,min}_{x \in X} \frac{1}{2}\|Ax - y^\delta\|^2 + \psi(\varphi_\Theta(x)) \qquad (21)$$

- $\varphi_\Theta : X \to Z$
- $\psi : Z \to [0, \infty]$ lower semi-continuous and coercive
- Regularizer $R_\Theta = \psi(\varphi_\Theta(x))$ is non-convex

Aim: Construct a regularizer R_Θ with small values for good x and large value for bad x.

Learned regularizers: NETT

NETT (Network Tikhonov)

$$T_\Theta(y^\delta) = \arg\min_{x \in X} \frac{1}{2}\|Ax - y^\delta\|^2 + \psi(\varphi_\Theta(x)) \tag{21}$$

- $\varphi_\Theta : X \to Z$
- $\psi : Z \to [0,\ \infty]$ lower semi-continuous and coercive
- Regularizer $R_\Theta = \psi(\varphi_\Theta(x))$ is non-convex

Aim: Construct a regularizer R_Θ with small values for good x and large value for bad x.

Learned regularizers: NETT

- $\varphi_\Theta(x)$ extracts artifacts from x

- $\psi(\cdot) = \|\cdot\|^2$

Learned regularizers: NETT

How to train $\varphi_\Theta(x)$?

- Take training pairs $\{(x_i^\dagger, 0)\}_{i=1}^N \cup \{(b_i, r_i)\}_{i=1}^N$
- $b_i = A^\dagger A x_i$ (images with artifacts)
- $r_i = x_i - A^\dagger A x_i$ (residual artifacts)
- Train network φ_Θ by minimizing

$$\hat{\Theta} = \arg\min \frac{1}{N} \sum \|\varphi_\Theta(x_i^\dagger)\|^2 + \frac{1}{N} \sum \|\varphi_\Theta(b_i) - r_i\|^2 \quad (22)$$

Learned regularizers: Adversarial

Given the training data $\{(y_i^\delta, x_i^\dagger)\}$ the aim is to find a regularizer R

$$R = \underset{R\in\mathcal{R}}{\arg\min} \sum_{i=1}^{N} \|\hat{x}_i - x_i^\dagger\| \tag{23}$$

s.t.

$$\hat{x}_i = \underset{x\in X}{\arg\min}\, \frac{1}{2}\|Ax - y_i^\delta\|^2 + R(x) \tag{24}$$

Approach: Train a neural network $R_\Theta : X \to \mathbb{R}$ to discriminate between the distributions $\mathbb{P}_n$ (good images) and $\mathbb{P}_r$ (bad images)

Learned regularizers: Adversarial

Given the training data $\{(y_i^\delta, x_i^\dagger)\}$ the aim is to find a regularizer R

$$R = \underset{R \in \mathcal{R}}{\arg\min} \sum_{i=1}^{N} \|\hat{x}_i - x_i^\dagger\| \tag{23}$$

s.t.

$$\hat{x}_i = \underset{x \in X}{\arg\min} \frac{1}{2}\|Ax - y_i^\delta\|^2 + R(x) \tag{24}$$

Approach: Train a neural network $R_\Theta : X \to \mathbb{R}$ to discriminate between the distributions $\mathbb{P}_n$ (good images) and $\mathbb{P}_r$ (bad images)

Recent approaches

- Learned methods

$$T_\Theta : Y \to X \tag{25}$$

- Learned regularizers

$$T_\Theta(y^\delta) = \arg\min_{x \in X} \ell(Ax,\ y^\delta) + R_\Theta(x) \tag{26}$$

- **Generative Networks**

$$T_\Theta(y^\delta) = \arg\min_{z \in Z} \ell(A\varphi_\Theta(z),\ y^\delta) + R_\Theta(z) \tag{27}$$

Generative networks

Consider a generative network $\varphi_\Theta(z)$ previously trained

- Θ is fixed after the training phase
- We can obtain images by sampling $\mathbf{z}$

For solving inverse problems (e.g.[9]):

$$\hat{z} = \arg\min_z \frac{1}{2}\|A\varphi_\Theta(z) - y^\delta\|^2 + R(z) \tag{28}$$

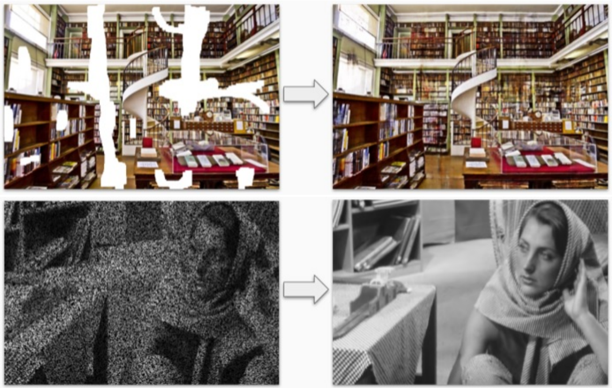
$$\hat{x} = \varphi_\Theta(\hat{z}) \tag{29}$$

[9] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G. Dimakis. "Compressed Sensing using Generative Models". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. 2017, pp. 537–546.

Section 3

Deep Image Prior

Examples [10]



[10] https://dmitryulyanov.github.io/deep_image_prior

Basic Idea[11]

Given measured noisy data

$$y^\delta = Ax^\dagger + \tau \tag{30}$$

1. Optimize a neural network $\varphi_\Theta(z_0)$ with a fixed input z_0

$$\hat{\Theta} = \arg\min_{\Theta} \frac{1}{2}\|A\varphi_\Theta(z_0) - y^\delta\|^2 \tag{31}$$

2. Set $\hat{x} = \varphi_{\hat{\Theta}}(z_0)$ as the reconstruction

[11] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. "Deep Image Prior". In: *CoRR* (2017). arXiv: 1711.10925.

Basic Idea[11]

Given measured noisy data

$$y^\delta = Ax^\dagger + \tau \tag{30}$$

1. Optimize a neural network $\varphi_\Theta(z_0)$ with a fixed input z_0

$$\hat{\Theta} = \arg\min_\Theta \frac{1}{2}\|A\varphi_\Theta(z_0) - y^\delta\|^2 \tag{31}$$

2. Set $\hat{x} = \varphi_{\hat{\Theta}}(z_0)$ as the reconstruction

[11] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. "Deep Image Prior". In: *CoRR* (2017). arXiv: 1711.10925.

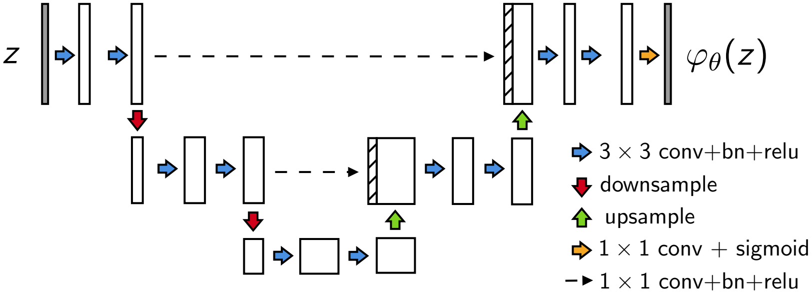
Some insights

- The network φ_Θ has a U-Net-like architecture
- It has enough expressive power to reproduce some noise
- Optimization method (ADAM[12]) with early stopping plays an important role
- Solving each instance requires training the network
- It takes a lot of time

[12] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

Task dependent hyper-parameters

- U-Net-like architecture
  - Number of scales (e.g. 2, 3, 4, 5, 6, ...)
  - Filter size per scale (e.g. 3, 5, ...)
  - Number of filters per scale (e.g. 8, 16, 32, 64, 128, ...)
  - Number of filters per skip connection (e.g. 2, 4, ...)



Section 4

Application in Computed Tomography

Radon transform

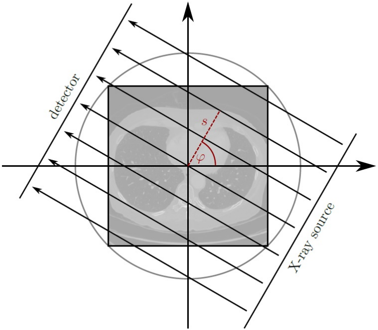


Figure: Parallel beam geometry

Example

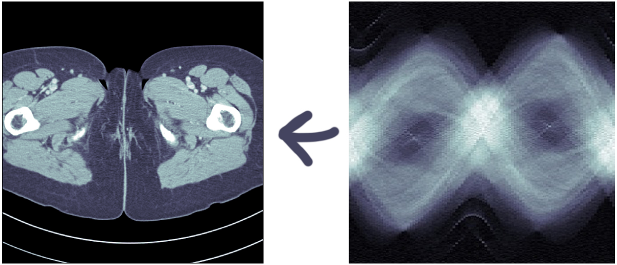
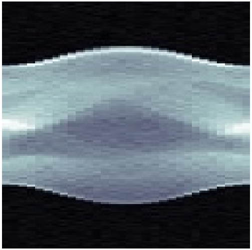


Figure: Human phantom and corresponding sinogram

Example a): Shepp-Logan phantom

- Parallel beam geometry (30 angles, 183 detectors)
- 5% white noise
- Visualization window: $[0.1, 0.4]$

(a) Ground truth (128×128)

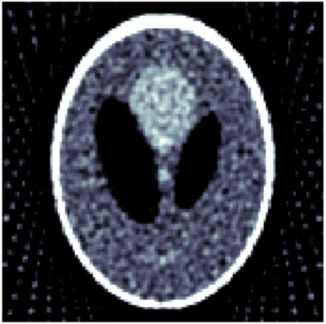


(b) Data (30×183)

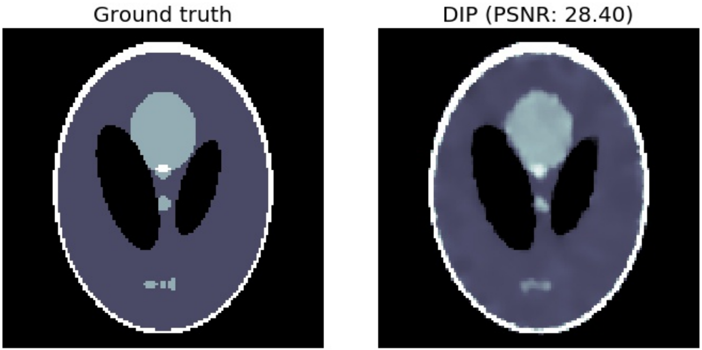
Example a): Shepp-Logan phantom

Ground truth

FBP (PSNR: 19.75)

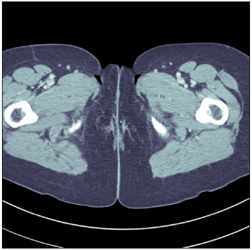


Example a): Shepp-Logan phantom

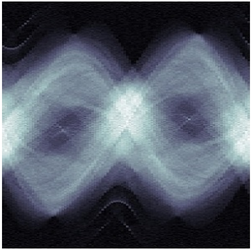


Example b): Human phantom[13]

- Case i: Fan-beam geometry (100 angles, 1000 detectors)
- Case ii: Fan-beam geometry (1000 angles, 1000 detectors)
- 5% white noise



(a) Ground truth (512×512)



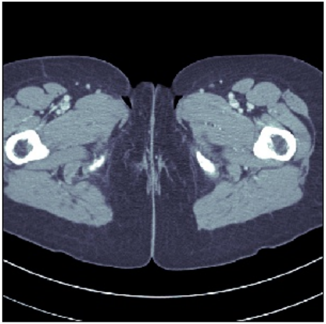
(b) Data (100×1000)

[13] Jonas Adler and Ozan Öktem. "Learned primal-dual reconstruction". In: *IEEE transactions on medical imaging* 37.6 (2018), pp. 1322–1332.

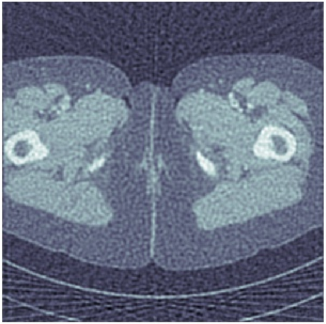
Example b): Human phantom

Case i: 100 angles

Ground truth

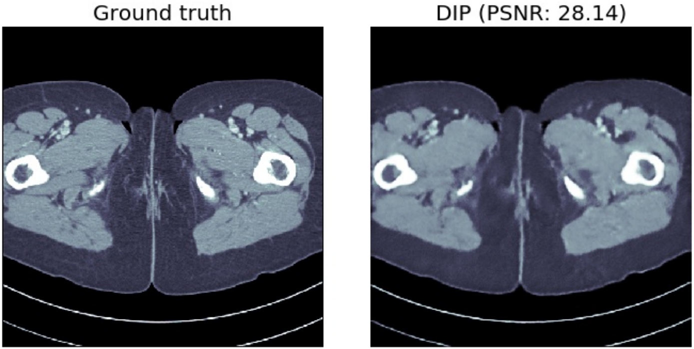


FBP (PSNR: 20.99)



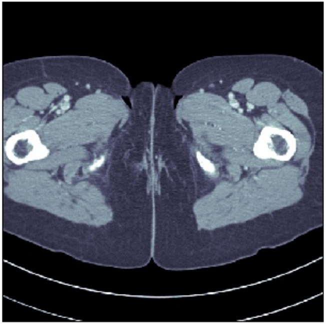
Example b): Human phantom

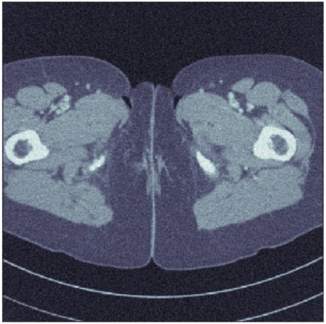
Case i: 100 angles



Example b): Human phantom

Case ii: 1000 angles





Example b): Human phantom

Case ii: 1000 angles

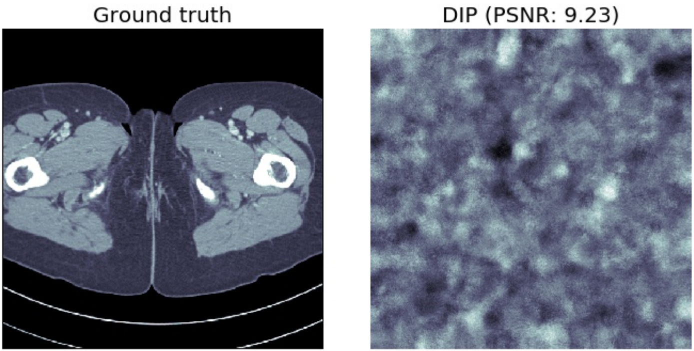


Figure: Iteration 0

Example b): Human phantom

Case ii: 1000 angles

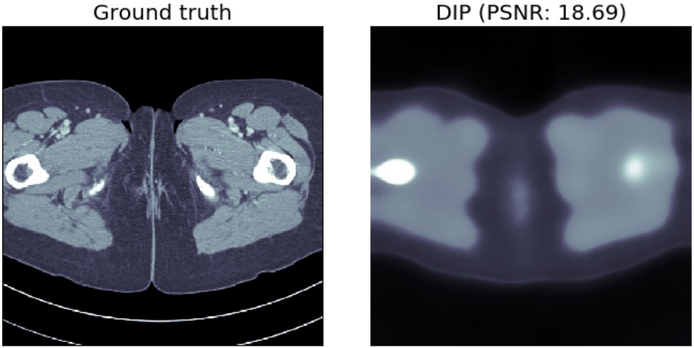


Figure: Iteration 100

Example b): Human phantom

Case ii: 1000 angles

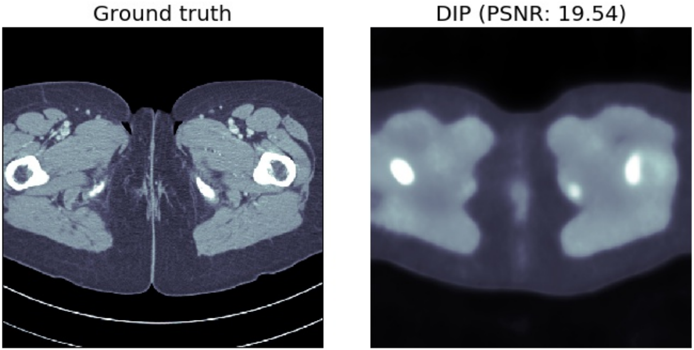


Figure: Iteration 200

Example b): Human phantom

Case ii: 1000 angles

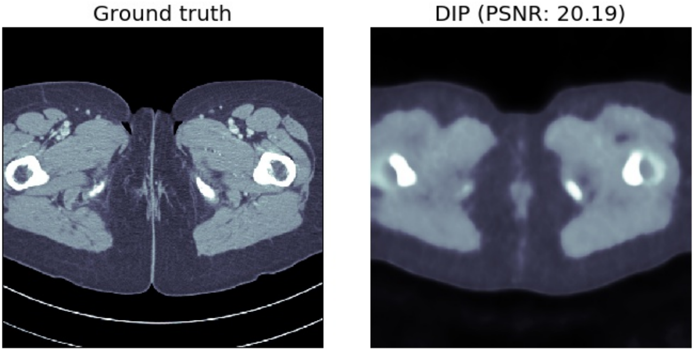


Figure: Iteration 300

Example b): Human phantom

Case ii: 1000 angles

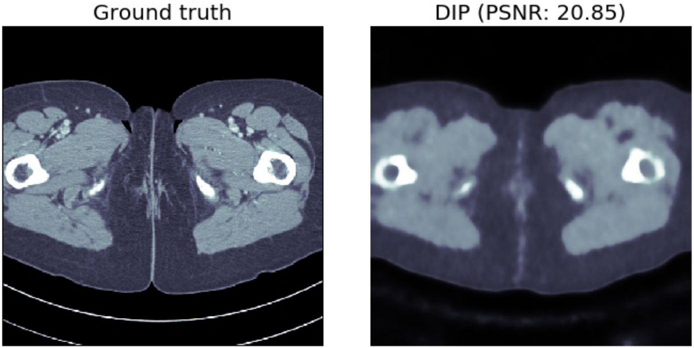


Figure: Iteration 400

Example b): Human phantom

Case ii: 1000 angles

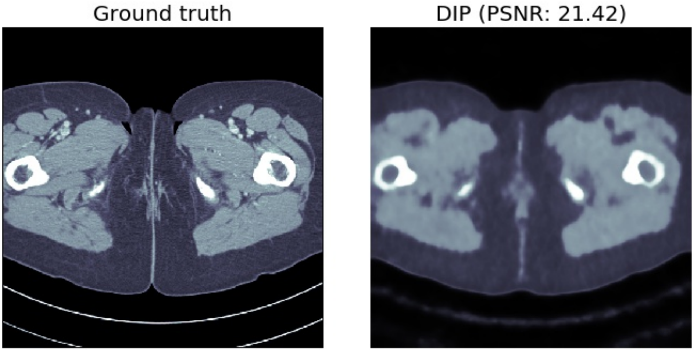


Figure: Iteration 500

Example b): Human phantom

Case ii: 1000 angles

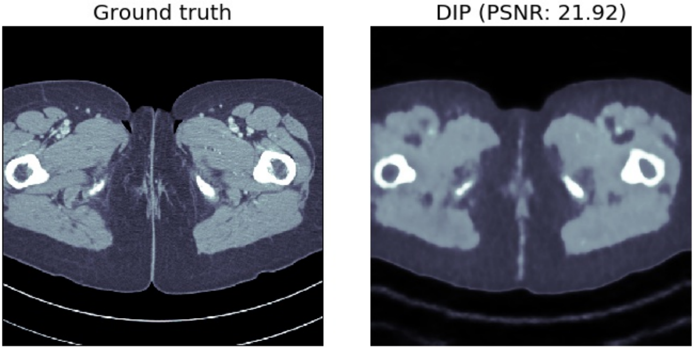


Figure: Iteration 600

Example b): Human phantom

Case ii: 1000 angles

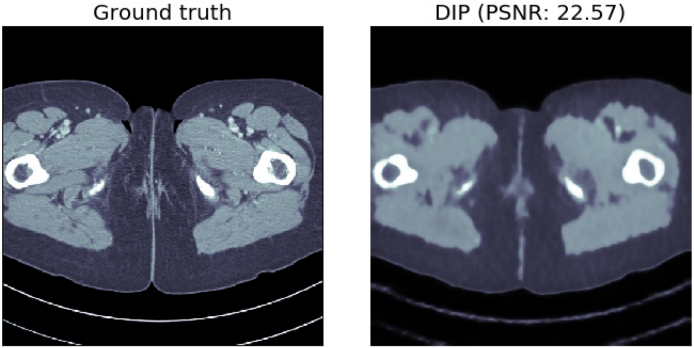


Figure: Iteration 700

Example b): Human phantom

Case ii: 1000 angles

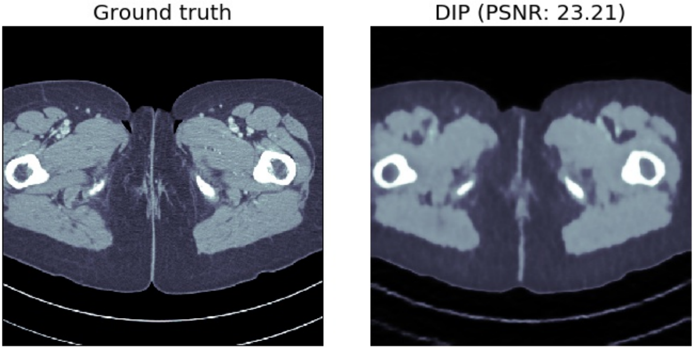


Figure: Iteration 800

Example b): Human phantom

Case ii: 1000 angles

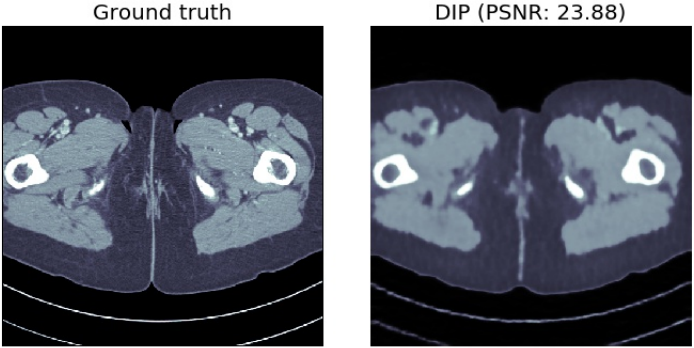


Figure: Iteration 900

Example b): Human phantom

Case ii: 1000 angles

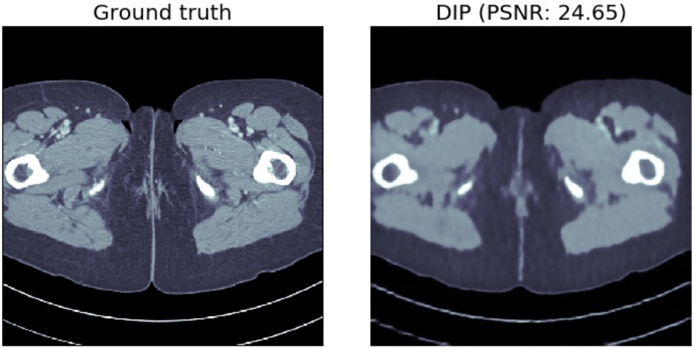


Figure: Iteration 1000

Example b): Human phantom

Case ii: 1000 angles

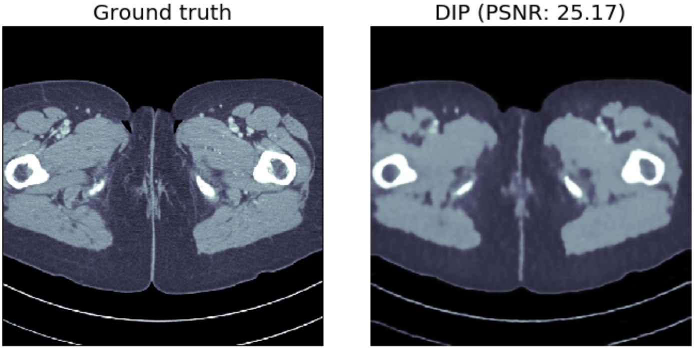


Figure: Iteration 1100

Example b): Human phantom

Case ii: 1000 angles

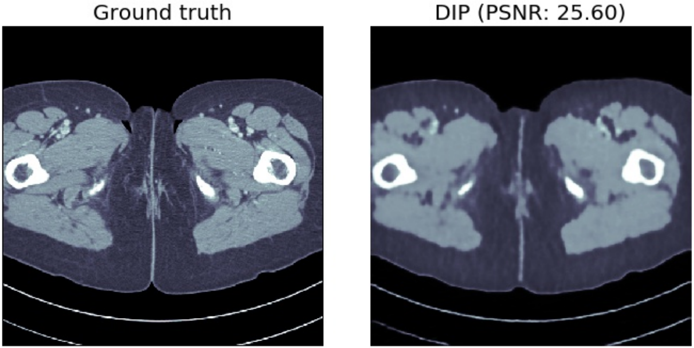


Figure: Iteration 1200

Example b): Human phantom

Case ii: 1000 angles

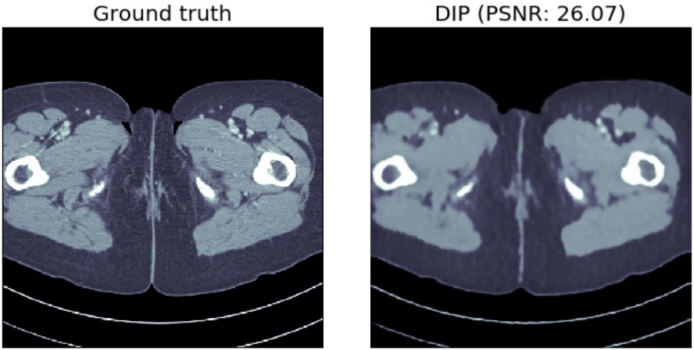


Figure: Iteration 1300

Example b): Human phantom

Case ii: 1000 angles

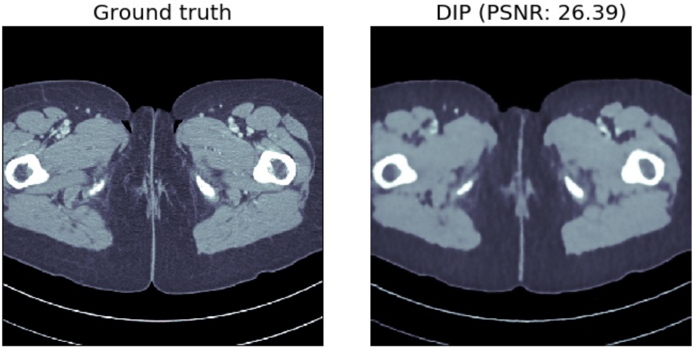


Figure: Iteration 1400

Example b): Human phantom

Case ii: 1000 angles

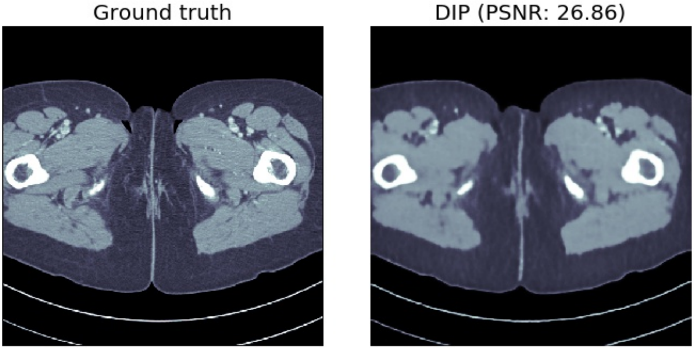


Figure: Iteration 1500

Example b): Human phantom

Case ii: 1000 angles

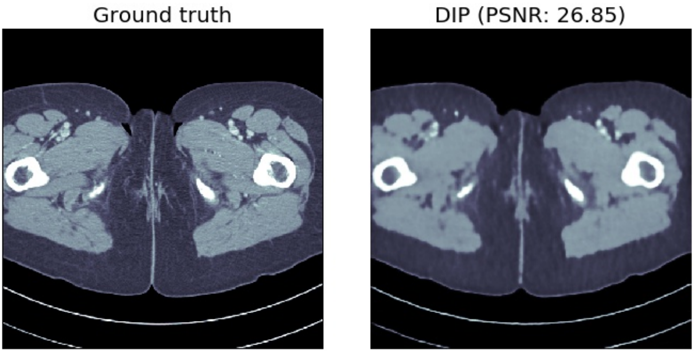


Figure: Iteration 1600

Example b): Human phantom

Case ii: 1000 angles

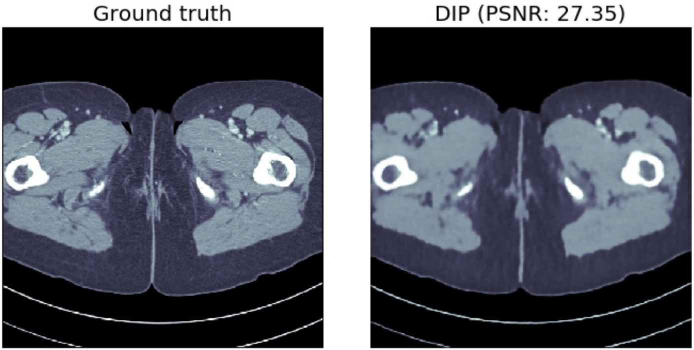


Figure: Iteration 1700

Example b): Human phantom

Case ii: 1000 angles

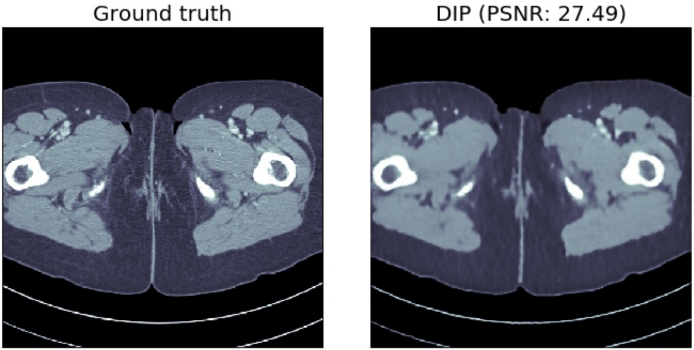


Figure: Iteration 1800

Example b): Human phantom

Case ii: 1000 angles

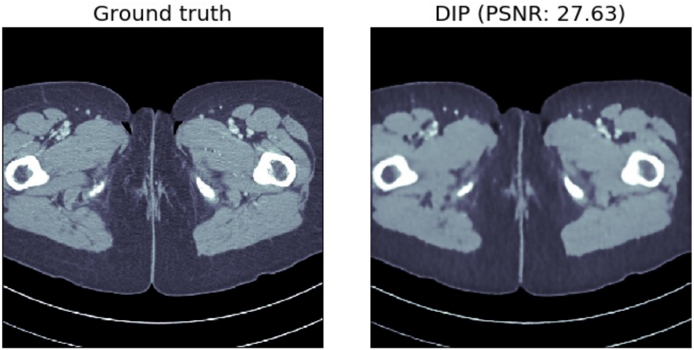


Figure: Iteration 1900

Example b): Human phantom

Case ii: 1000 angles

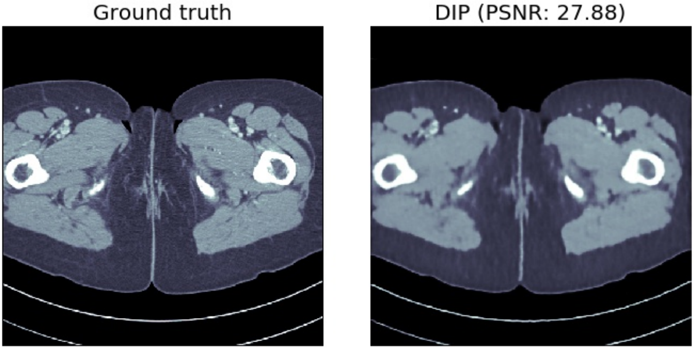


Figure: Iteration 2000

Example b): Human phantom

Case ii: 1000 angles

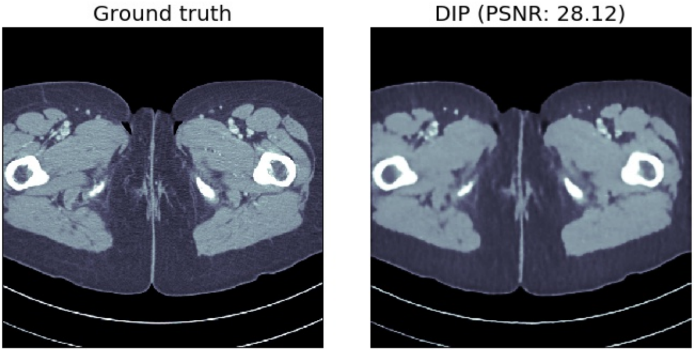


Figure: Iteration 2100

Example b): Human phantom

Case ii: 1000 angles

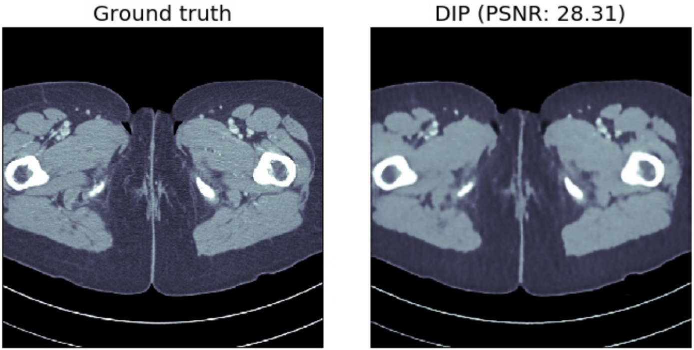


Figure: Iteration 2200

Example b): Human phantom

Case ii: 1000 angles

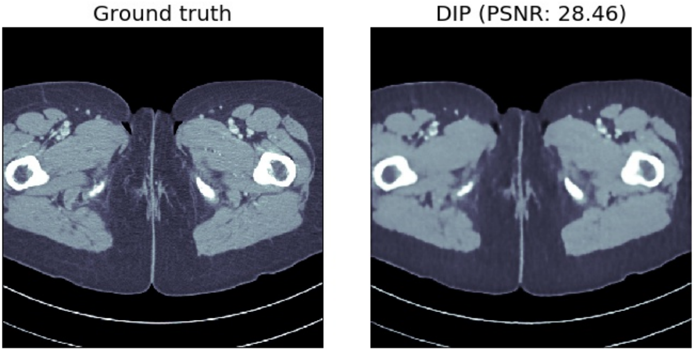


Figure: Iteration 2300

Example b): Human phantom

Case ii: 1000 angles

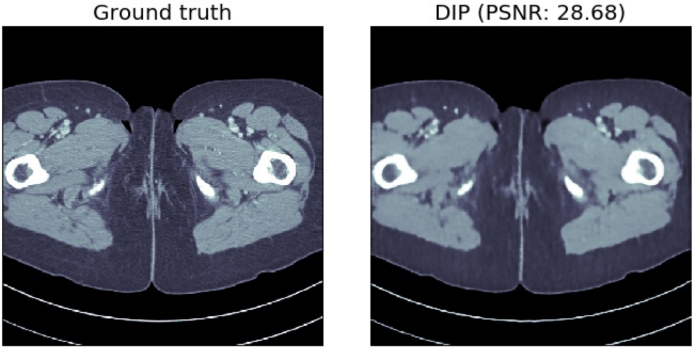
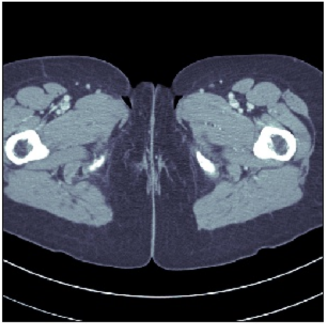


Figure: Iteration 2400

Example b): Human phantom

Case ii: 1000 angles



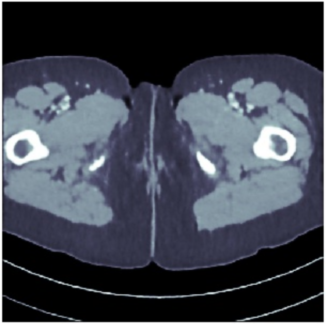


Figure: Iteration 2500

Example b): Human phantom

Case ii: 1000 angles

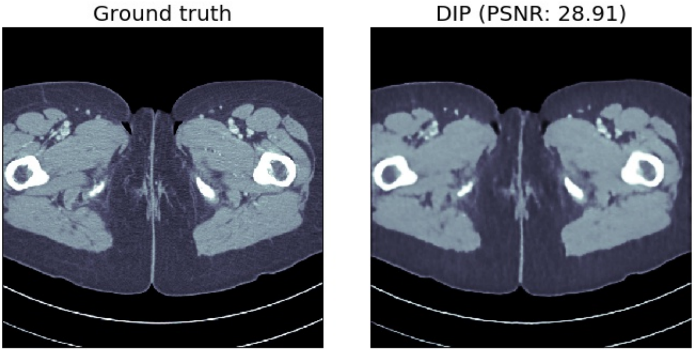


Figure: Iteration 2600

Example b): Human phantom

Case ii: 1000 angles

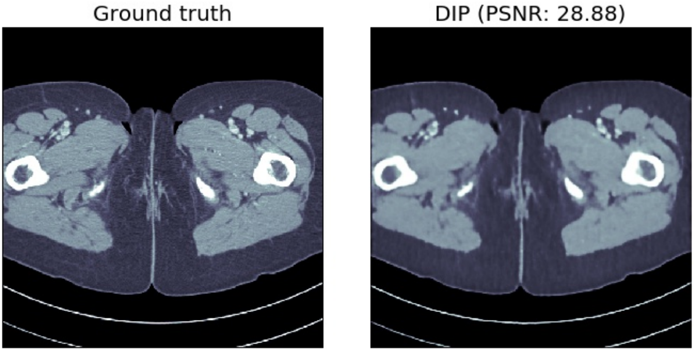


Figure: Iteration 2700

Example b): Human phantom

Case ii: 1000 angles

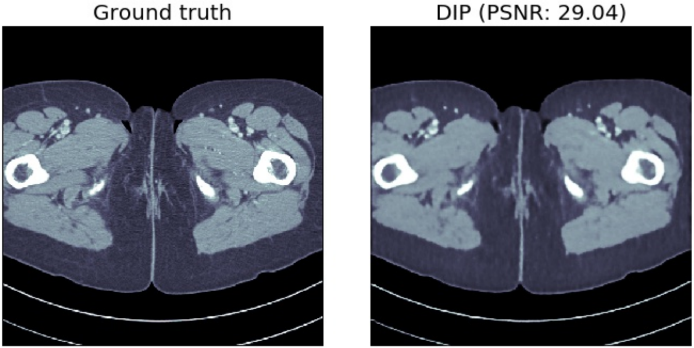


Figure: Iteration 2800

Example b): Human phantom

Case ii: 1000 angles

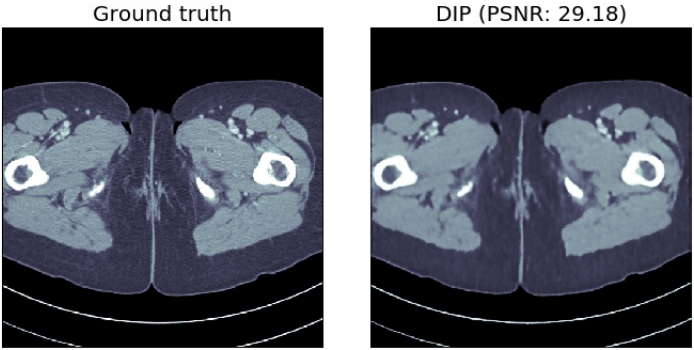


Figure: Iteration 2900

Example b): Human phantom

Case ii: 1000 angles

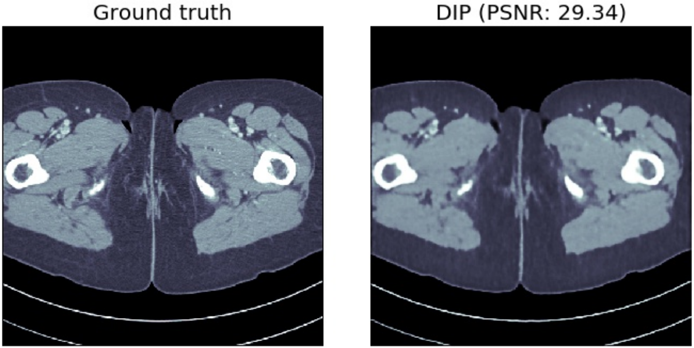


Figure: Iteration 3000

Example b): Human phantom

Case ii: 1000 angles

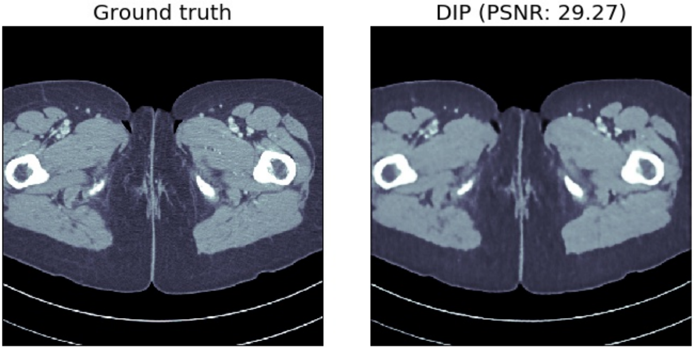


Figure: Iteration 3100

Example b): Human phantom

Case ii: 1000 angles

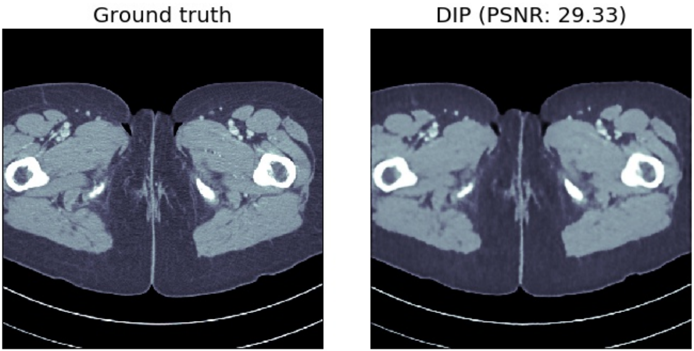


Figure: Iteration 3200

Example b): Human phantom

Case ii: 1000 angles

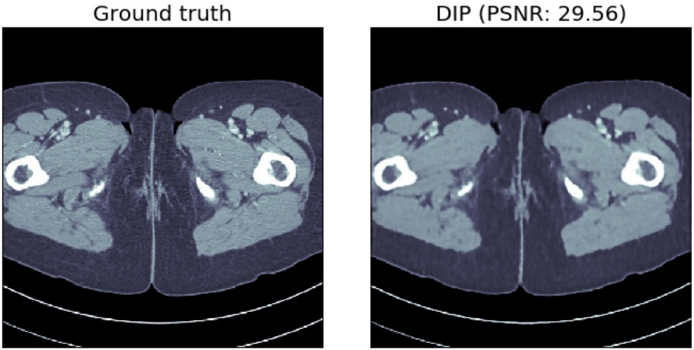


Figure: Iteration 3300

Example b): Human phantom

Case ii: 1000 angles

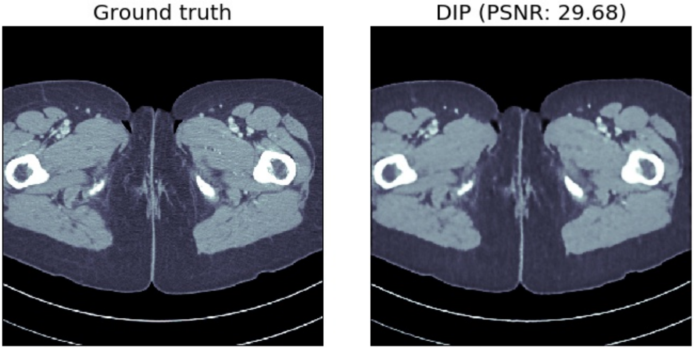


Figure: Iteration 3400

Example b): Human phantom

Case ii: 1000 angles

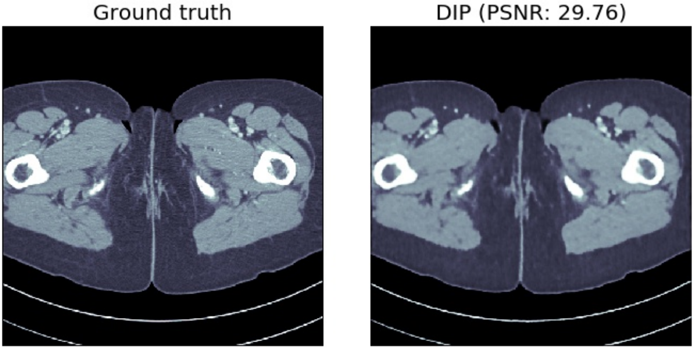


Figure: Iteration 3500

Example b): Human phantom

Case ii: 1000 angles

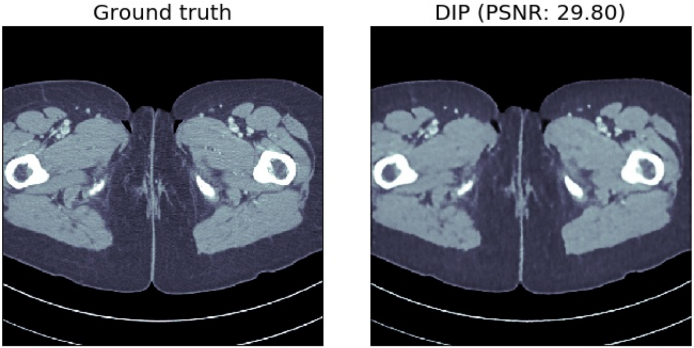


Figure: Iteration 3600

Example b): Human phantom

Case ii: 1000 angles

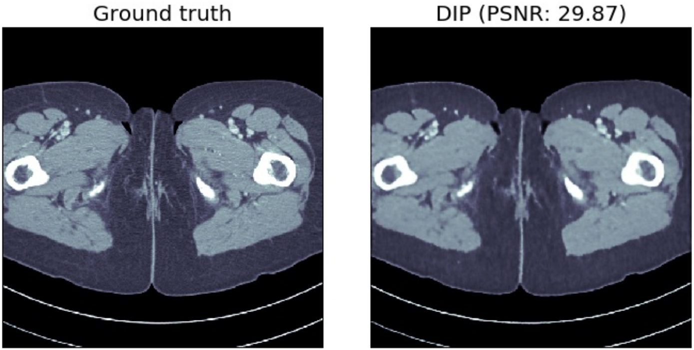


Figure: Iteration 3700

Example b): Human phantom

Case ii: 1000 angles

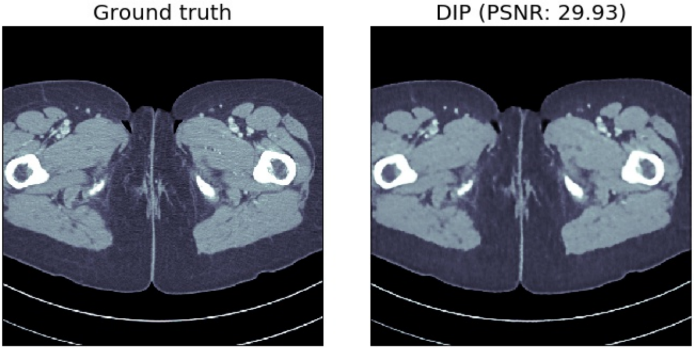


Figure: Iteration 3800

Example b): Human phantom

Case ii: 1000 angles

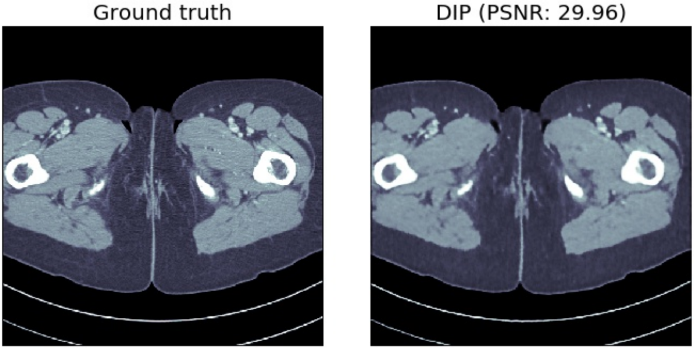


Figure: Iteration 3900

Example b): Human phantom

Case ii: 1000 angles

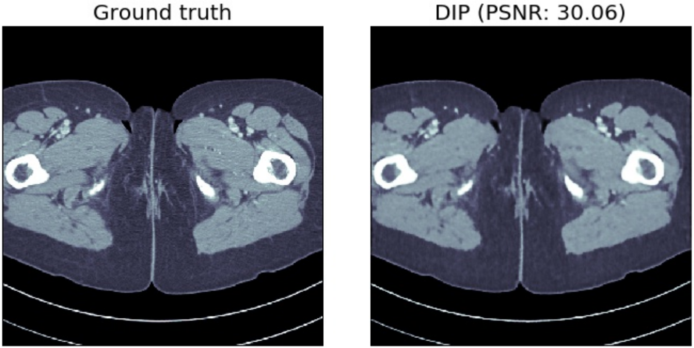


Figure: Iteration 4000

Example b): Human phantom

Case ii: 1000 angles

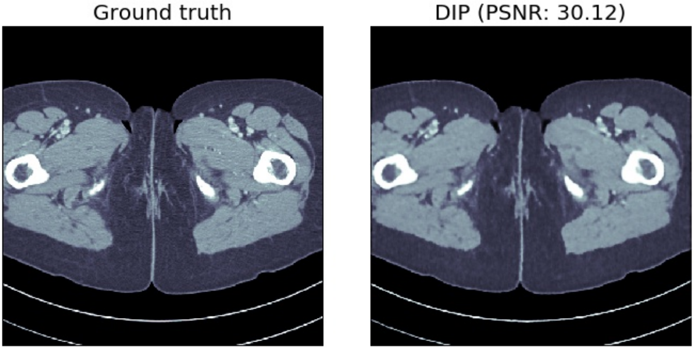


Figure: Iteration 4100

Example b): Human phantom

Case ii: 1000 angles

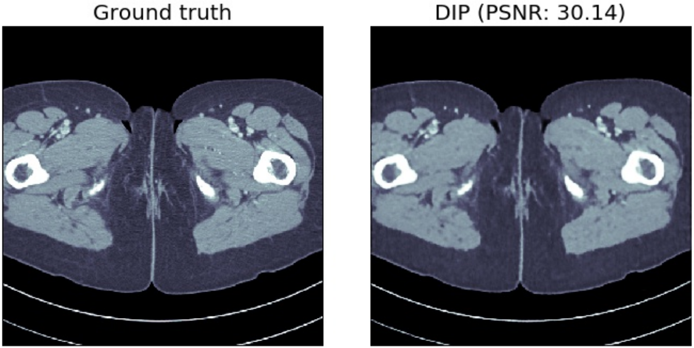


Figure: Iteration 4200

Example b): Human phantom

Case ii: 1000 angles

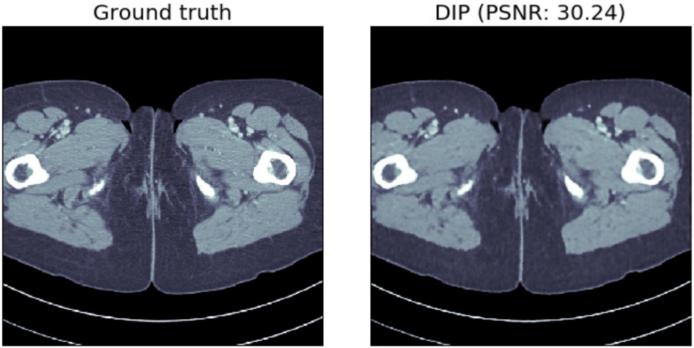


Figure: Iteration 4300

Example b): Human phantom

Case ii: 1000 angles

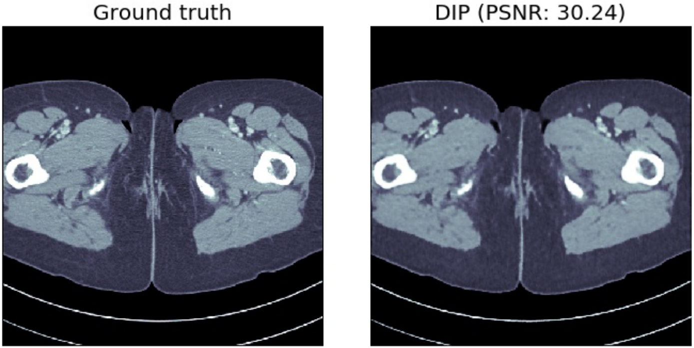


Figure: Iteration 4400

Example b): Human phantom

Case ii: 1000 angles

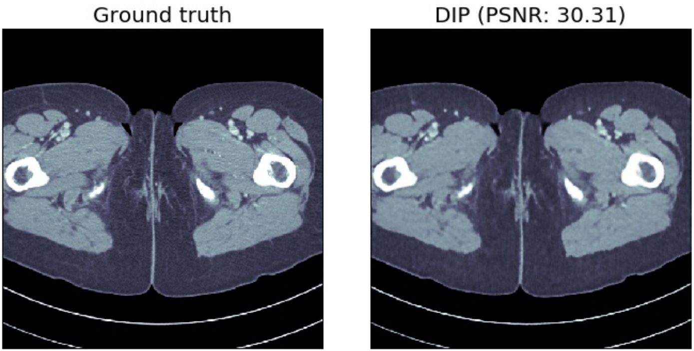


Figure: Iteration 4500

Example b): Human phantom

Case ii: 1000 angles

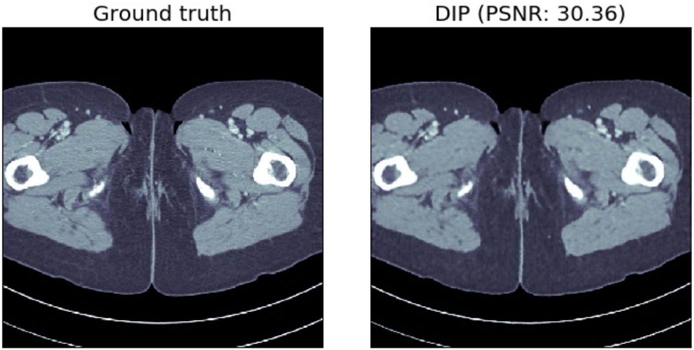


Figure: Iteration 4600

Example b): Human phantom

Case ii: 1000 angles

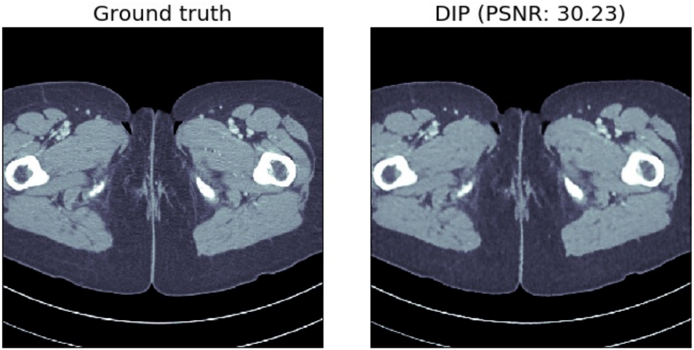


Figure: Iteration 4700

Example b): Human phantom

Case ii: 1000 angles

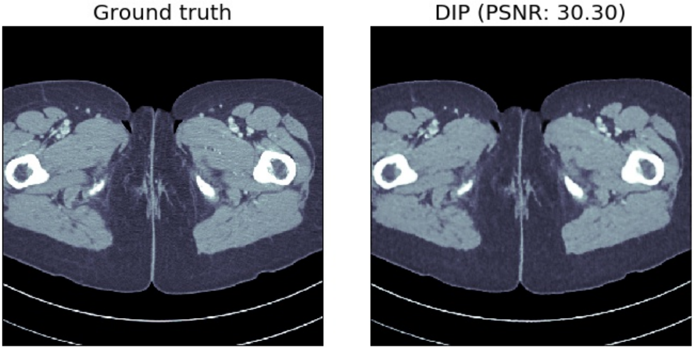


Figure: Iteration 4800

Example b): Human phantom

Case ii: 1000 angles

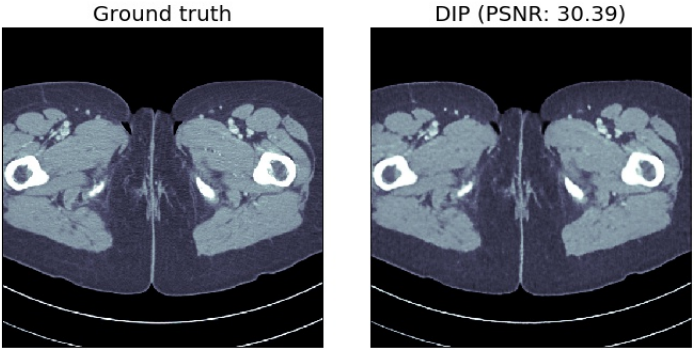


Figure: Iteration 4900

Example b): Human phantom

Case ii: 1000 angles

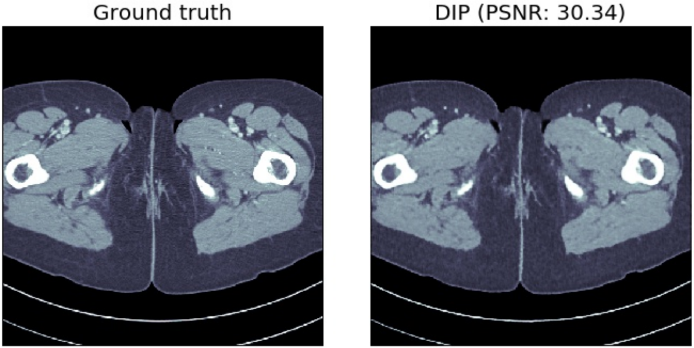


Figure: Iteration 5000

Example b): Human phantom

Case ii: 1000 angles

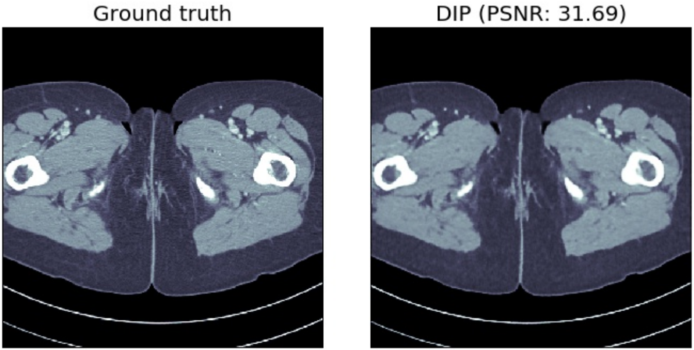


Figure: Final result

Example b): Human phantom

Case ii: 1000 angles (Running time $\approx$ 7 min)

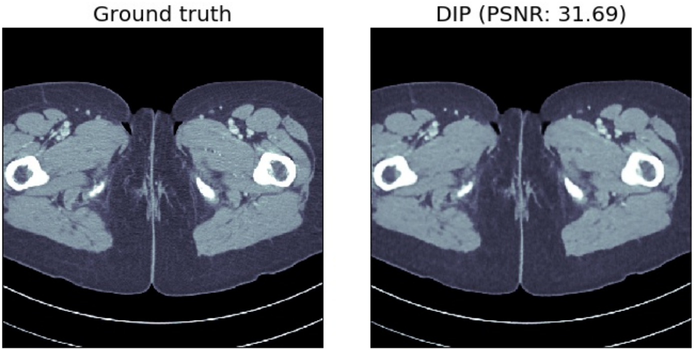


Figure: Final result

Implementation

Libraries:

- DIP source code[14]
- Operator Discretization Library (ODL)[15]

Training parameters:

- Iterations: 5000
- Learning rate: 10^{-3}
- Regularization noise: 10^{-2}

Architecture:

- Number of scales: 5
- Filter size per scale: 3
- Number of filters per scale: 128
- Number of filters per skip connection: 4

Hardware:

- Nvidia GeForce GTX 1080

[14] https://github.com/DmitryUlyanov/deep-image-prior

[15] https://github.com/odlgroup/odl

Section 5

Analytic Deep Prior

Simple architecture

Can the DIP approach be used to solve ill-posed inverse problems?

Consider a trivial network $\varphi_\Theta(z) = \Theta$

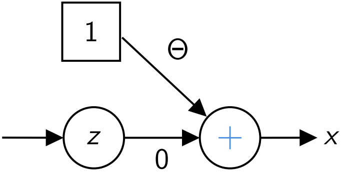
$\implies$ Minimizing $\|A\varphi_\Theta(z) - y^\delta\|^2 = \|A\Theta - y^\delta\|^2$ by gradient descent with respect to Θ is equivalent to the classical Landweber iteration

$$\alpha \sim \frac{1}{n} \tag{32}$$

Simple architecture

Can the DIP approach be used to solve ill-posed inverse problems?

Consider a trivial network $\varphi_\Theta(z) = \Theta$



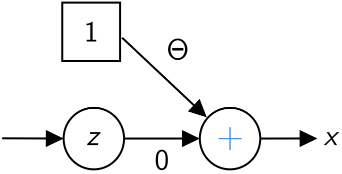
$\implies$ Minimizing $\|A\varphi_\Theta(z) - y^\delta\|^2 = \|A\Theta - y^\delta\|^2$ by gradient descent with respect to Θ is equivalent to the classical Landweber iteration

$$\alpha \sim \frac{1}{n} \tag{32}$$

Simple architecture

Can the DIP approach be used to solve ill-posed inverse problems?

Consider a trivial network $\varphi_\Theta(z) = \Theta$



$\implies$ Minimizing $\|A\varphi_\Theta(z) - y^\delta\|^2 = \|A\Theta - y^\delta\|^2$ by gradient descent with respect to Θ is equivalent to the classical Landweber iteration

$$\alpha \sim \frac{1}{n} \tag{32}$$

Unrolled proximal gradient architecture

Consider a fully connected feed-forward network with L layers

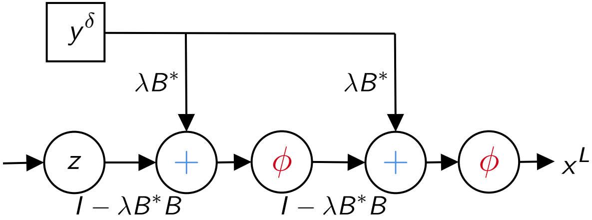
$$\varphi_\Theta(z) = x^L, \tag{33}$$

where

$$x^{k+1} = \phi\left(Wx^k + b\right) \tag{34}$$

- The affine linear map $\Theta = (W, b)$ is the same for all layers
- The matrix W is restricted to obey $I - W = \lambda B^* B$ for any B and the bias is determined via $b = \lambda B^* y^\delta$
- The activation function of the network is chosen as the proximal mapping of a regularizing functional $\lambda\alpha R : X \to \mathbb{R}$

Unrolled proximal gradient architecture



$\implies$ $\varphi_\Theta(z) = x^L$ is identical to the L-th iterate of a proximal gradient descent method for minimizing

$$J_B(x) = \frac{1}{2}\|Bx - y^\delta\|^2 + \alpha R(x) \tag{35}$$

Deep priors and Tikhonov functionals

Given: measured data $y^\delta \in Y$, fixed $\alpha > 0$, convex penalty functional $R : X \to \mathbb{R}$ and the operator $A \in \mathcal{L}(X, Y)$

Solve:

$$\hat{B} = \operatorname*{arg\,min}_{B \in \mathcal{L}(X,Y)} \underbrace{\frac{1}{2}\|Ax(B) - y^\delta\|^2}_{F(B)} \tag{36}$$

subject to

$$x(B) = \operatorname*{arg\,min}_{x \in X} \frac{1}{2}\|Bx - y^\delta\|^2 + \alpha R(x) \tag{37}$$

Result: $x(\hat{B})$ as the solution to the inverse problem

$\implies$ **Analytic Deep Prior**

Deep priors and Tikhonov functionals

Given: measured data $y^\delta \in Y$, fixed $\alpha > 0$, convex penalty functional $R : X \to \mathbb{R}$ and the operator $A \in \mathcal{L}(X, Y)$

Solve:

$$\hat{B} = \operatorname*{arg\,min}_{B \in \mathcal{L}(X,Y)} \underbrace{\frac{1}{2} \| A x(B) - y^\delta \|^2}_{F(B)} \tag{36}$$

subject to

$$x(B) = \operatorname*{arg\,min}_{x \in X} \frac{1}{2} \| Bx - y^\delta \|^2 + \alpha R(x) \tag{37}$$

Result: $x(\hat{B})$ as the solution to the inverse problem

$\Longrightarrow$ **Analytic Deep Prior**

Deep priors and Tikhonov functionals

Given: measured data $y^\delta \in Y$, fixed $\alpha > 0$, convex penalty functional $R : X \to \mathbb{R}$ and the operator $A \in \mathcal{L}(X, Y)$

Solve:

$$\hat{B} = \underset{B \in \mathcal{L}(X,Y)}{\arg\min} \underbrace{\frac{1}{2}\|Ax(B) - y^\delta\|^2}_{F(B)} \tag{36}$$

subject to

$$x(B) = \underset{x \in X}{\arg\min} \frac{1}{2}\|Bx - y^\delta\|^2 + \alpha R(x) \tag{37}$$

Result: $x(\hat{B})$ as the solution to the inverse problem

$\Longrightarrow$ **Analytic Deep Prior**

Deep priors and Tikhonov functionals

Given: measured data $y^\delta \in Y$, fixed $\alpha > 0$, convex penalty functional $R : X \to \mathbb{R}$ and the operator $A \in \mathcal{L}(X, Y)$

Solve:

$$\hat{B} = \underset{B \in \mathcal{L}(X,Y)}{\arg\min} \underbrace{\frac{1}{2}\|Ax(B) - y^\delta\|^2}_{F(B)} \tag{36}$$

subject to

$$x(B) = \underset{x \in X}{\arg\min} \frac{1}{2}\|Bx - y^\delta\|^2 + \alpha R(x) \tag{37}$$

Result: $x(\hat{B})$ as the solution to the inverse problem

$\Longrightarrow$ **Analytic Deep Prior**

Deep priors and Tikhonov functionals

Theorem

Let

$$\psi(x, B) = \operatorname*{Prox}_{\lambda\alpha R}\left(x - \lambda B^*(Bx - y^\delta)\right) - x \tag{38}$$

then

$$\partial F(B) = \partial x(B)^* A^*(Ax(B) - y^\delta) \tag{39}$$

with

$$\partial x(B) = -\psi_x(x(B), B)^{-1}\psi_B(x(B), B) \tag{40}$$

This yields the gradient descent iteration

$$B^{\ell+1} = B^\ell - \eta\partial F(B^\ell). \tag{41}$$

Example i)

Assume $R(x) = \frac{1}{2}\|x\|^2$

Simple case: $x^\dagger = u$, where u is a singular function of A $(Au = \sigma v)$

$$y^\delta = Au + \delta v = (\sigma + \delta)v \tag{42}$$

A lengthy computation exploiting $B^0 = A$ and the iteration $B^{\ell+1} = B^\ell - \eta \partial F(B^\ell)$ yields

$$B^{\ell+1} = B^\ell - c_\ell v u^* \tag{43}$$

with

$$c_\ell = \eta\sigma(\sigma+\delta)^2(\alpha + \beta_\ell^2 - \sigma\beta_\ell)\frac{\beta_\ell^2 - \alpha}{(\beta_\ell^2 + \alpha)^3}$$

β_ℓ: singular value of B^ℓ $(B^\ell u = \beta^\ell v)$

Example i)

Assume $R(x) = \frac{1}{2}\|x\|^2$

Simple case: $x^\dagger = u$, where u is a singular function of A $(Au = \sigma v)$

$$y^\delta = Au + \delta v = (\sigma + \delta)v \tag{42}$$

A lengthy computation exploiting $B^0 = A$ and the iteration $B^{\ell+1} = B^\ell - \eta\partial F(B^\ell)$ yields

$$B^{\ell+1} = B^\ell - c_\ell v u^* \tag{43}$$

with

$$c_\ell = \eta\sigma(\sigma+\delta)^2(\alpha + \beta_\ell^2 - \sigma\beta_\ell)\frac{\beta_\ell^2 - \alpha}{(\beta_\ell^2 + \alpha)^3}$$

β_ℓ: singular value of B^ℓ $(B^\ell u = \beta^\ell v)$

Example i)

Assume $R(x) = \frac{1}{2}\|x\|^2$

Simple case: $x^\dagger = u$, where u is a singular function of A $(Au = \sigma v)$

$$y^\delta = Au + \delta v = (\sigma + \delta)v \tag{42}$$

A lengthy computation exploiting $B^0 = A$ and the iteration $B^{\ell+1} = B^\ell - \eta \partial F(B^\ell)$ yields

$$B^{\ell+1} = B^\ell - c_\ell v u^* \tag{43}$$

with

$$c_\ell = \eta\sigma(\sigma+\delta)^2(\alpha + \beta_\ell^2 - \sigma\beta_\ell)\frac{\beta_\ell^2 - \alpha}{(\beta_\ell^2 + \alpha)^3}$$

β_ℓ: singular value of B^ℓ $(B^\ell u = \beta^\ell v)$

Example i)

This results in the sequence β^ℓ converging to

$$\beta_\infty = \begin{cases} \frac{\sigma}{2} \pm \sqrt{\frac{\sigma^2}{4} - \alpha} & \sigma \geq 2\sqrt{\alpha} \\ \sqrt{\alpha} & \sigma < 2\sqrt{\alpha} \end{cases} \tag{44}$$

and the sequence $x(B^\ell)$ with the unique attractive stationary point[16]

$$x(B^\infty) = \begin{cases} \frac{1}{\sigma}(\sigma + \delta)u & \sigma \geq 2\sqrt{\alpha} \\ \frac{1}{2\sqrt{\alpha}}(\sigma + \delta)u & \sigma < 2\sqrt{\alpha} \end{cases} \tag{45}$$

[16] Sören Dittmer, Tobias Kluth, Peter Maass, and Daniel Otero Baguer. "Regularization by architecture: A deep prior approach for inverse problems". In: *CoRR* abs/1812.03889 (2018). arXiv: 1812.03889. URL: http://arxiv.org/abs/1812.03889.

Example ii)

Assume $R(x) = \frac{1}{2}\|x\|^2$ and we optimize over

$$B \in \left\{ \tilde{B} \in \mathcal{L}(X,Y) \;\middle|\; \tilde{B} = \sum_i \beta_i v_i u_i^*,\ \beta_i \in \mathbb{R}_+ \cup \{0\} \right\} \tag{46}$$

where $\{u_i, \sigma_i, v_i\}$ is the singular value decomposition of A

Theorem

There exist a global minimizer given by $B_\alpha = \sum \beta_i^\alpha v_i u_i^*$ with

$$\beta_i^\alpha = \begin{cases} \frac{\sigma_i}{2} + \sqrt{\frac{\sigma_i^2}{4} - \alpha} & \sigma_i \geq 2\sqrt{\alpha} \\ \sqrt{\alpha} & \sigma_i < 2\sqrt{\alpha} \end{cases} \tag{47}$$

Example ii)

Assume $R(x) = \frac{1}{2}\|x\|^2$ and we optimize over

$$B \in \left\{ \tilde{B} \in \mathcal{L}(X, Y) \;\middle|\; \tilde{B} = \sum_i \beta_i v_i u_i^*,\ \beta_i \in \mathbb{R}_+ \cup \{0\} \right\} \tag{46}$$

where $\{u_i, \sigma_i, v_i\}$ is the singular value decomposition of A

Theorem

There exist a global minimizer given by $B_\alpha = \sum \beta_i^\alpha v_i u_i^*$ *with*

$$\beta_i^\alpha = \begin{cases} \frac{\sigma_i}{2} + \sqrt{\frac{\sigma_i^2}{4} - \alpha} & \sigma_i \geq 2\sqrt{\alpha} \\ \sqrt{\alpha} & \sigma_i < 2\sqrt{\alpha} \end{cases} \tag{47}$$

Example ii)

Recap the regularized pseudo inverse in terms of filter functions:

$$T_\alpha(y^\delta) = \sum_{\sigma_i > 0} F_\alpha(\sigma_i) \frac{1}{\sigma_i} \langle y^\delta, v_i \rangle u_i \tag{48}$$

Theorem (Soft TSVD)

The regularized pseudo inverse $K_\alpha(y^\delta) = x(B_\alpha, y^\delta)$ is an order optimal regularization method[17] *given by the filter function*

$$F_\alpha(\sigma) = \begin{cases} 1 & \sigma \geq 2\sqrt{\alpha} \\ \frac{\sigma}{2\sqrt{\alpha}} & \sigma < 2\sqrt{\alpha} \end{cases} \tag{49}$$

[17] Alfred Karl Louis. *Inverse und schlecht gestellte Probleme.* Wiesbaden: Vieweg+Teubner Verlag, 1989.

Example ii)

Recap the regularized pseudo inverse in terms of filter functions:

$$T_\alpha(y^\delta) = \sum_{\sigma_i > 0} F_\alpha(\sigma_i) \frac{1}{\sigma_i} \langle y^\delta, v_i \rangle u_i \qquad (48)$$

Theorem (Soft TSVD)

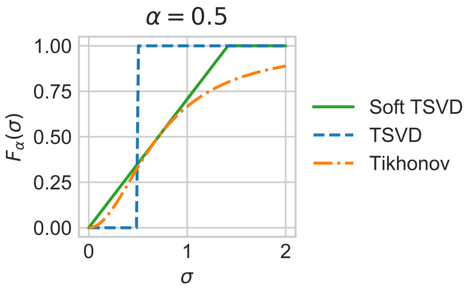
The regularized pseudo inverse $K_\alpha(y^\delta) = x(B_\alpha, y^\delta)$ *is an order optimal regularization method*[17] *given by the filter function*

$$F_\alpha(\sigma) = \begin{cases} 1 & \sigma \geq 2\sqrt{\alpha} \\ \frac{\sigma}{2\sqrt{\alpha}} & \sigma < 2\sqrt{\alpha} \end{cases} \qquad (49)$$

[17] Alfred Karl Louis. *Inverse und schlecht gestellte Probleme*. Wiesbaden: Vieweg+Teubner Verlag, 1989.

Example ii)

Comparison with other regularization methods



Thanks!