Deep Learning for Inverse Problems

Where are we? How far can we go?

Jonas Adler[1, 2] Ozan Öktem[1]

[1]Department of Mathematics
KTH - Royal Institute of Technology, Stockholm, Sweden

[2]Research and Physics
Elekta, Stockholm, Sweden

Inverse Problems

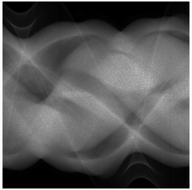
$$y = \mathcal{A}(x^*) + e.$$

$y \in Y$	Data
$x^* \in X$	Image
$\mathcal{A} : X \to Y$	Forward operator
$e \in Y$	Noise

Inverse Problems

$$y = \mathcal{A}(x^*) + e.$$

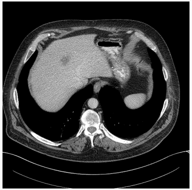
$y \in Y$	Data
$x^* \in X$	Image
$\mathcal{A} : X \to Y$	Forward operator
$e \in Y$	Noise

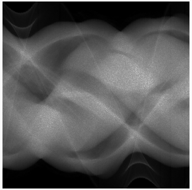


Inverse Problems

$$y = \mathcal{A}(x^*) + e.$$

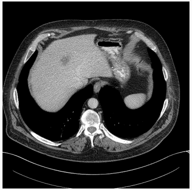
$y \in Y$	Data
$x^* \in X$	Image
$\mathcal{A} : X \to Y$	Forward operator
$e \in Y$	Noise



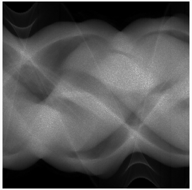


$$y = \mathcal{A}(x^*) + e.$$

$y \in Y$	Data
$x^* \in X$	Image
$\mathcal{A} : X \to Y$	Forward operator
$e \in Y$	Noise



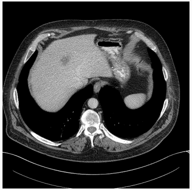
$\xrightarrow{\mathcal{A}}$



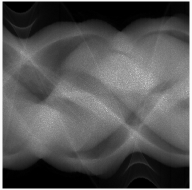
Inverse Problems

$$y = \mathcal{A}(x^*) + e.$$

$y \in Y$	Data
$x^* \in X$	Image
$\mathcal{A} : X \to Y$	Forward operator
$e \in Y$	Noise



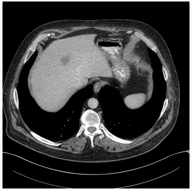
$\xrightarrow{\mathcal{A}}$



Inverse Problems

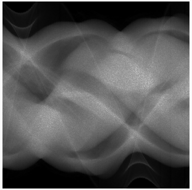
$$y = \mathcal{A}(x^*) + e.$$

$y \in Y$	Data
$x^* \in X$	Image
$\mathcal{A} : X \to Y$	Forward operator
$e \in Y$	Noise



$$\xrightarrow{\mathcal{A}}$$

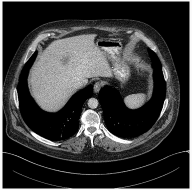
$$\underset{\text{"}\,\mathcal{A}^{-1}\,\text{"}}{\longleftarrow}$$



Inverse Problems

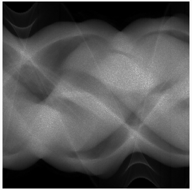
$$y = \mathcal{A}(x^*) + e.$$

$y \in Y$	Data
$x^* \in X$	Image
$\mathcal{A} : X \to Y$	Forward operator
$e \in Y$	Noise



$$\xrightarrow{\mathcal{A}}$$

$$\underset{\text{"}\,\mathcal{A}^{-1}\,\text{"}}{\longleftarrow}$$



The problem is ill-posed: non-uniqueness, instability

Statistical viewpoint

Data $y \in Y$ is a single observation generated by Y-valued random variable $\mathbf{y}$ where

$$\mathbf{y} = \mathcal{A}(\mathbf{x}) + \mathbf{e}.$$

Statistical viewpoint

Data $y \in Y$ is a single observation generated by Y-valued random variable $\mathbf{y}$ where

$$\mathbf{y} = \mathcal{A}(\mathbf{x}) + \mathbf{e}.$$

Full solution: A probability distribution on model parameter space X

$$\mathbb{P}(\mathbf{x} \mid \mathbf{y} = y)$$

This is a full characterization of the reconstruction, including uncertainty.

Statistical viewpoint

Data $y \in Y$ is a single observation generated by Y-valued random variable $\mathbf{y}$ where

$$\mathbf{y} = \mathcal{A}(\mathbf{x}) + \mathbf{e}.$$

Full solution: A probability distribution on model parameter space X

$$\mathbb{P}(\mathbf{x} \mid \mathbf{y} = y)$$

This is a full characterization of the reconstruction, including uncertainty.

Typical solution: Compute some estimator, e.g. the conditional mean

$$\mathbb{E}\big[\mathbf{x} \mid \mathbf{y} = y\big]$$

Risk minimization

Theorem (Conditional Mean)
Assume that Y is a measurable metric space, X a measurable Hilbert space, and $\mathbf{y}$ and $\mathbf{x}$ are Y- and X-valued random variables, respectively. Let

$$h^* = \operatorname*{arg\,min}_{h\colon Y \to X} \mathbb{E}\Big[\big\| h(\mathbf{y}) - \mathbf{x} \big\|_X^2\Big].$$

Then $h^*(y) := \mathbb{E}\big[\mathbf{x} \mid \mathbf{y} = y\big]$ almost everywhere.

Empirical risk minimization

- Suppose we aim to compute

$$\mathbb{E}\big[\mathbf{x} \mid \mathbf{y} = y\big]$$

- This can be done by solving

Empirical risk minimization

- Suppose we aim to compute

$$\mathbb{E}\big[\mathbf{x} \mid \mathbf{y} = y\big]$$

- This can be done by solving

$$\min_{h\colon Y \to X} \mathbb{E}\Big[\big\| h(\mathbf{y}) - \mathbf{x} \big\|_X^2\Big].$$

Empirical risk minimization

- Suppose we aim to compute

$$\mathbb{E}\left[\mathbf{x} \mid \mathbf{y} = y\right]$$

- This can be done by solving

$$\min_{h\colon Y \to X} \mathbb{E}\left[\left\| h(\mathbf{y}) - \mathbf{x} \right\|_X^2\right].$$

The minimization is over all measurable functions

Empirical risk minimization

- Suppose we aim to compute
$$\mathbb{E}\big[\mathbf{x} \mid \mathbf{y} = y\big]$$
- This can be done by solving
$$\min_{\theta \in \Theta} \mathbb{E}\Big[\big\|\mathcal{A}_\theta^\dagger(\mathbf{y}) - \mathbf{x}\big\|_X^2\Big].$$

The minimization is over all measurable functions

Restrict minimization to some tractable subset

Empirical risk minimization

- Suppose we aim to compute

$$\mathbb{E}\big[\mathbf{x} \mid \mathbf{y} = y\big]$$

- This can be done by solving

$$\min_{\theta \in \Theta} \mathbb{E}\Big[\big\|\mathcal{A}^\dagger_\theta(\mathbf{y}) - \mathbf{x}\big\|_X^2\Big].$$

Expectation is taken over the unknown joint distribution

Empirical risk minimization

- Suppose we aim to compute

$$\mathbb{E}\big[\mathbf{x} \mid \mathbf{y} = y\big]$$

- This can be done by solving

$$\min_{\theta\in\Theta} \frac{1}{N}\sum_{i=1}^{N} \big\|\mathcal{A}_\theta^\dagger(y_i) - x_i\big\|_X^2$$

Expectation is taken over the unknown joint distribution

Replace with empirical mean

Empirical risk minimization

- Suppose we aim to compute

$$\mathbb{E}\big[\mathbf{x} \mid \mathbf{y} = y\big]$$

- This can be done by solving

$$\min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} \big\| \mathcal{A}_\theta^\dagger(y_i) - x_i \big\|_X^2$$

This is a "computationally tractable" formulation, we just need to pick $\{\mathcal{A}_\theta^\dagger\}_{\theta \in \Theta}$.

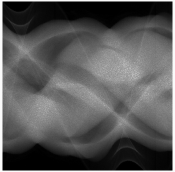
Learned inversion methods

Architecture: Specification of the class of operators $\{\mathcal{A}_\theta^\dagger\}_{\theta \in \Theta}$.

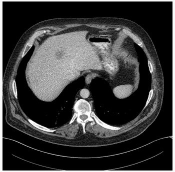
Learned inversion methods

Architecture: Specification of the class of operators $\{\mathcal{A}_\theta^\dagger\}_{\theta \in \Theta}$.

Main complication: $\mathcal{A}_\theta^\dagger : Y \to X$.

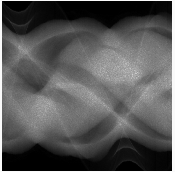


$\xrightarrow[\mathcal{A}_\theta^\dagger]{}$

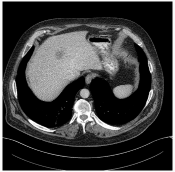


Architecture: Specification of the class of operators $\{\mathcal{A}_\theta^\dagger\}_{\theta \in \Theta}$.

Main complication: $\mathcal{A}_\theta^\dagger : Y \to X$.



$\xrightarrow[\mathcal{A}_\theta^\dagger]{}$

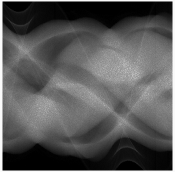


- Fully learned: Learn everything, disregard structure.

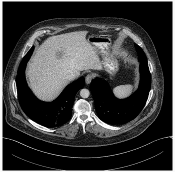
Learned inversion methods

Architecture: Specification of the class of operators $\{\mathcal{A}_\theta^\dagger\}_{\theta \in \Theta}$.

Main complication: $\mathcal{A}_\theta^\dagger : Y \to X$.



$\xrightarrow[\mathcal{A}_\theta^\dagger]{}$

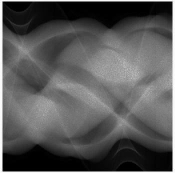


- Fully learned: Learn everything, disregard structure.
- Learned post-processing: First apply standard inverse, then denoise $\mathcal{A}_\theta^\dagger = P_\theta \circ \mathcal{A}^\dagger$

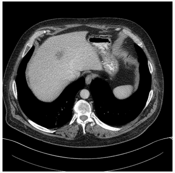
Learned inversion methods

Architecture: Specification of the class of operators $\{\mathcal{A}_\theta^\dagger\}_{\theta \in \Theta}$.

Main complication: $\mathcal{A}_\theta^\dagger : Y \to X$.



$\underset{\mathcal{A}_\theta^\dagger}{\longrightarrow}$



- Fully learned: Learn everything, disregard structure.
- Learned post-processing: First apply standard inverse, then denoise $\mathcal{A}_\theta^\dagger = P_\theta \circ \mathcal{A}^\dagger$
- Learned iterative schemes: Embed physics inside deep neural network

Quality of reconstruction

How well does this actually work?

Measure *generalization gap*:

$$\mathbb{E}\left[\left\|\mathcal{A}^{\dagger}_{\theta^*}(\mathbf{y}) - \mathbf{x}\right\|^2_X\right] - \mathbb{E}\left[\left\|\mathbb{E}[\mathbf{x} \mid \mathbf{y}] - \mathbf{x}\right\|^2_X\right].$$

Results on toy case

Results for ray transform inversion in 2D:

- Inverse problem:
$$y = \mathcal{A}(x) + e$$
- Geometry: Parallel beam, sparse view (30 angles)
- Noise: 5% additive Gaussian
- Training data: 128×128 pixel ellipses

Results on toy case

Results for ray transform inversion in 2D:

- Inverse problem:

$$y = \mathcal{A}(x) + e$$

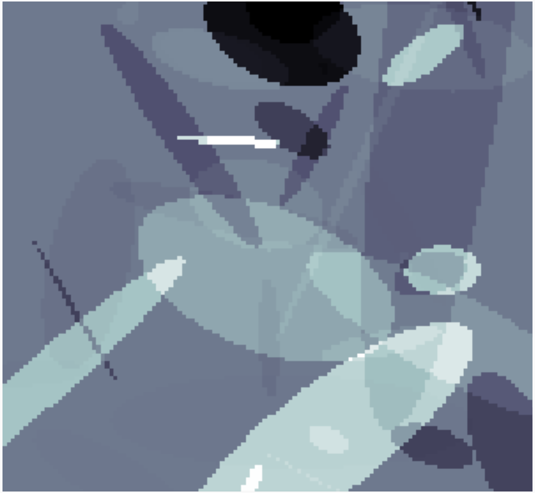
- Geometry: Parallel beam, sparse view (30 angles)
- Noise: 5% additive Gaussian
- Training data: 128×128 pixel ellipses

Compare to:

- FBP
- Total Variation
- Post-processing deep learning by U-Net
- Conditional expectation, $\mathbb{E}(\mathbf{x} \mid y)$, via MCMC

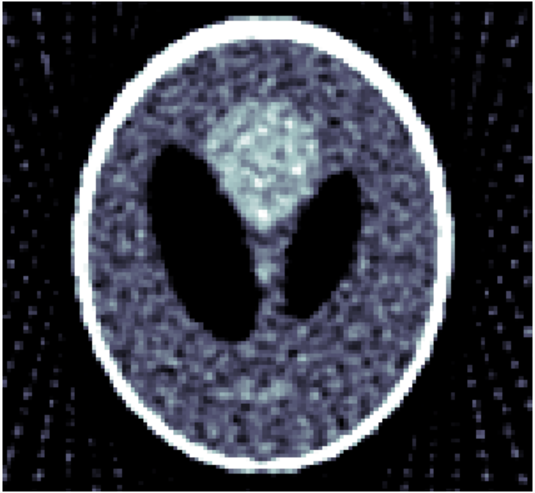
Measure relative error:

$$\frac{\mathbb{E}\left[\left\|\mathcal{A}^{\dagger}_{\theta^*}(\mathbf{y}) - \mathbf{x}\right\|_X^2\right]}{\mathbb{E}\left[\left\|\mathbb{E}\left[\mathbf{x} \mid \mathbf{y}\right] - \mathbf{x}\right\|_X^2\right]}$$



Training data

Phantom



FBP

Normalized error: 372

Phantom

TV

Normalized error: 56.0

Phantom

Learned Post-processing
Normalized error: 42.2

Phantom

Learned Iterative

Normalized error: 5.2

Phantom

Conditional Expectation

Normalized error: 1

Conclusions on learned reconstruction

- We can find a reconstruction operator by solving a minimization problem
- Architecture: Specification of the class of operators $\{\mathcal{A}_\theta^\dagger\}_{\theta \in \Theta}$.
- Learning:

$$\min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} \left\| \mathcal{A}_\theta^\dagger(y_i) - x_i \right\|_X^2$$

- Empirically, current methods are remarkably close to optimal

What now?

- Apparently deep learning techniques are great for the conditional mean
- What about other estimators?
- Maximum a-posteriori is very hard

What now?

- Apparently deep learning techniques are great for the conditional mean
- What about other estimators?
- Maximum a-posteriori is very hard
- But, what about finding the whole posterior?

Generative Advesarial Networks

- Main idea: train two networks, generator G and discriminator D
- Generator tries to generate "true" samples, discriminator tries to say "good/bad"

Generative Advesarial Networks

- Main idea: train two networks, generator G and discriminator D
- Generator tries to generate "true" samples, discriminator tries to say "good/bad"

Conditional Wasserstein GAN

Input: Supervised training data (x_i, y_i) generated by $(\mathbf{x}, \mathbf{y})$.

Goal: Sample from unknown posterior $\mathbb{P}(\mathbf{x} \mid \mathbf{y} = y)$.

Approach: Learn how to sample from posterior by solving

$$\min_{\theta} \mathbb{E}_{\mathbf{y} \sim \mathbb{P}_{data}} \Big[\mathcal{W}\big(\mathrm{G}_\theta(\mathbf{y}), \mathbb{P}(\mathbf{x} \mid \mathbf{y}) \big) \Big].$$

We minimize the *Wasserstein* distance between the random variables $\mathrm{G}_\theta(\mathbf{y})$ and $\mathbb{P}(\mathbf{x} \mid \mathbf{y})$!

Conditional Wasserstein GAN

Input: Supervised training data (x_i, y_i) generated by $(\mathbf{x}, \mathbf{y})$.

Goal: Sample from unknown posterior $\mathbb{P}(\mathbf{x} \mid \mathbf{y} = y)$.

Approach: Learn how to sample from posterior by solving

$$\min_{\theta} \mathbb{E}_{\mathbf{y} \sim \mathbb{P}_{data}} \Big[\mathcal{W}\big(\mathrm{G}_\theta(\mathbf{y}), \mathbb{P}(\mathbf{x} \mid \mathbf{y})\big) \Big].$$

Re-write using the Kantorovich-Rubinstein dual characterization of $\mathcal{W}$.

Conditional Wasserstein GAN

Input: Supervised training data (x_i, y_i) generated by $(\mathbf{x}, \mathbf{y})$.

Goal: Sample from unknown posterior $\mathbb{P}(\mathbf{x} \mid \mathbf{y} = y)$.

Approach: Learn how to sample from posterior by solving

$$\min_{\theta} \left\{ \max_{\mathrm{D} \in Lip(X)} \mathbb{E}\left[\mathrm{D}(\mathbf{x}, \mathbf{y}) - \mathrm{D}(\mathrm{G}_\theta(\mathbf{y})) \right] \right\}.$$

Re-write using the Kantorovich-Rubinstein dual characterization of $\mathcal{W}$.

Conditional Wasserstein GAN

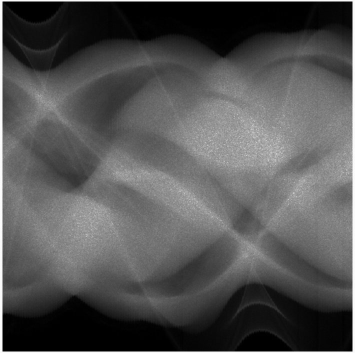
Input: Supervised training data (x_i, y_i) generated by $(\mathbf{x}, \mathbf{y})$.

Goal: Sample from unknown posterior $\mathbb{P}(\mathbf{x} \mid \mathbf{y} = y)$.

Approach: Learn how to sample from posterior by solving

$$\min_{\theta} \left\{ \max_{\mathrm{D} \in Lip(X)} \mathbb{E}\left[\mathrm{D}(\mathbf{x}, \mathbf{y}) - \mathrm{D}(\mathrm{G}_\theta(\mathbf{y})) \right] \right\}.$$

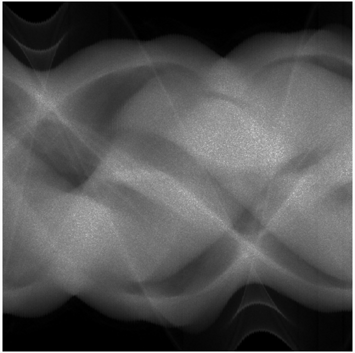
Formulation useful for deep learning



Data

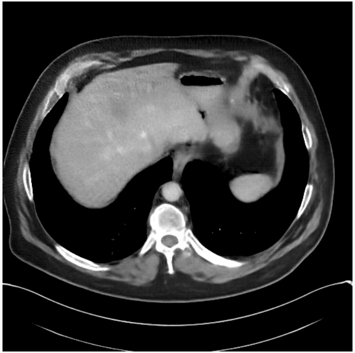
FBP

- Case: Patient with suspected metastasis to the liver.
- Data: Clinical helical 3D CT data, 2% of a normal dose (ultra low-dose).



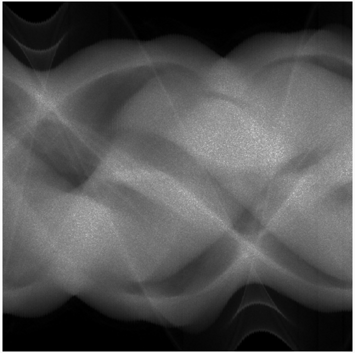
Data

FBP



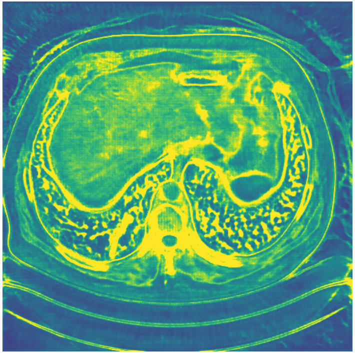
Posterior mean

- Case: Patient with suspected metastasis to the liver.
- Data: Clinical helical 3D CT data, 2% of a normal dose (ultra low-dose).



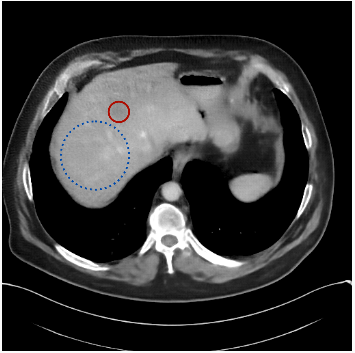
Data

FBP



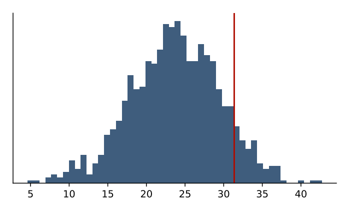
Standard deviation

- Case: Patient with suspected metastasis to the liver.
- Data: Clinical helical 3D CT data, 2% of a normal dose (ultra low-dose).

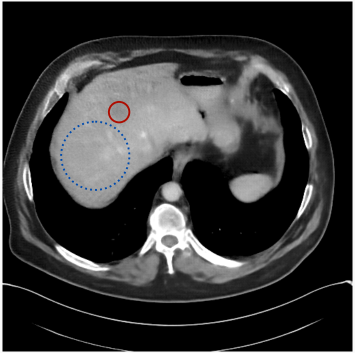


Posterior mean with ROI

- Case: Patient with suspected metastasis to the liver.
- Data: Clinical helical 3D CT data, 2% of a normal dose (ultra low-dose).
- Liver lesion: $\triangle =$ difference in average contrast between ROI and liver.
- Hypothesis test: Based on 1 000 samples, the ROI contains a lesion at 95%

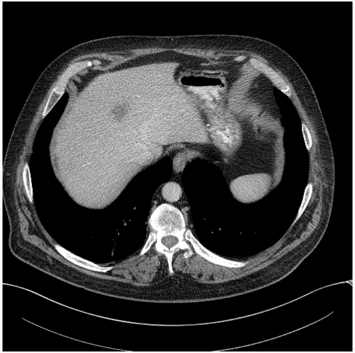


Histogram of $\triangle$

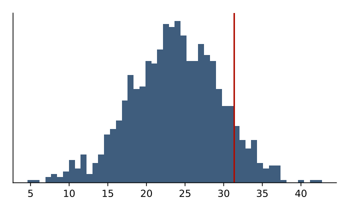


Posterior mean with ROI

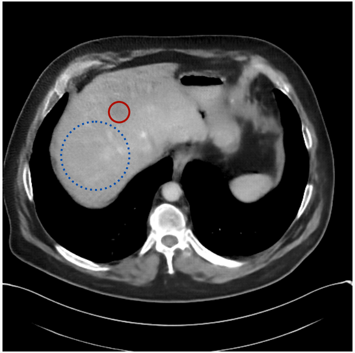
- Case: Patient with suspected metastasis to the liver.
- Data: Clinical helical 3D CT data, 2% of a normal dose (ultra low-dose).
- Liver lesion: $\triangle$ = difference in average contrast between ROI and liver.
- Hypothesis test: Based on 1 000 samples, the ROI contains a lesion at 95%



Normal dose image



Histogram of $\triangle$



Posterior mean with ROI

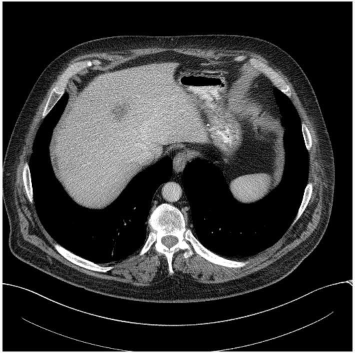
- Case: Patient with suspected metastasis to the liver.
- Data: Clinical helical 3D CT data, 2% of a normal dose (ultra low-dose).
- Liver lesion: $\triangle =$ difference in average contrast between ROI and liver.
- Hypothesis test: Based on 1 000 samples, the ROI contains a lesion at 95%

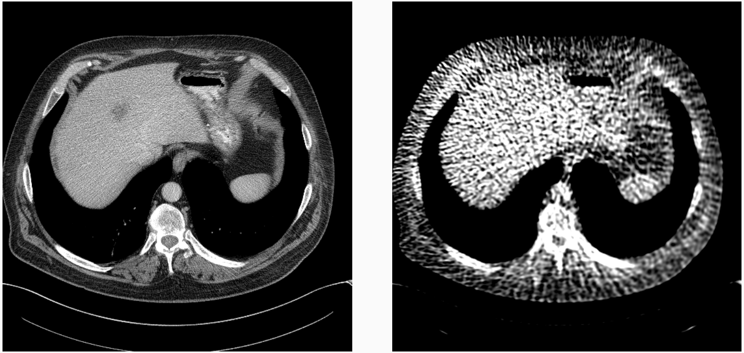
Conclusion

- Deep Learning methods for inverse problems building on empirical risk minimization are very powerful
- Fruitful ways forward involve questioning what we're trying to compute
- Posterior sampling is one such option

Postdoc in Deep Learning based Reconstruction for Spectral-CT

- Theory and methods for machine learning in image reconstruction.
- We've got the worlds first clinical photon counting spectral-CT data.
- Very nice position (great group, travel, salary)
- Pursued jointly with MedTechLabs and the Medical Imaging group at KTH.



Thank you for your attention!