Deep Bayesian Inversion

Jonas Adler

Department of Mathematics
KTH - Royal Institute of Technology, Stockholm, Sweden

Research and Physics
Elekta, Stockholm, Sweden

Overview

- Bayesian Inversion
- Direct Estimation
- Posterior Sampling

Bayesian Inversion

Bayesian Inversion

Inverse problem (Statistical viewpoint)

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

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

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. We don't need to select estimators (e.g. task adapted becomes irrelevant).

Bayesian Inversion: Theory

Recall some nice properties from yesterday:

- The posterior almost always exists
- The mapping
$$y \to \mathbb{P}(\mathbf{x} \mid \mathbf{y} = y)$$
is continuous.
- We can characterize convergence (Bernstein-von Mises)

Bayesian Inversion: Classical methods

- Bayes Law:

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

- We know the data likelihood

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

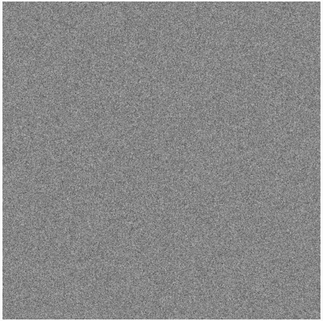
- Only have to specify the prior

$$\mathbb{P}(x)$$

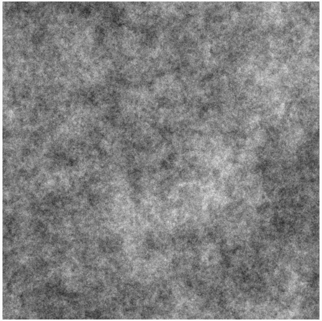
- Standard approach: Gibbs priors

$$\mathbb{P}(x) = e^{-S(x)}$$

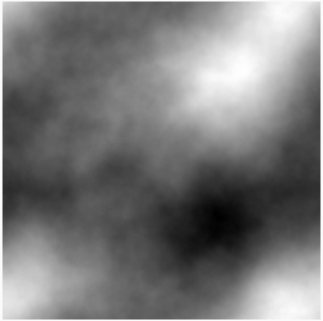
Bayesian Inversion: Samples



$S(x) = \|x\|_2^2$

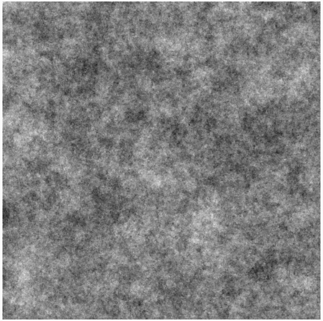


$S(x) = \|\nabla x\|_2^2$

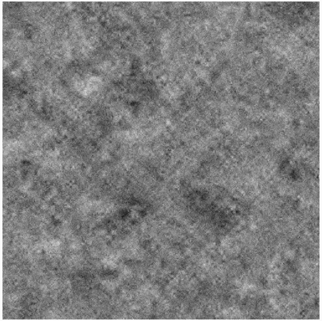


$S(x) = \|\Delta x\|_2^2$

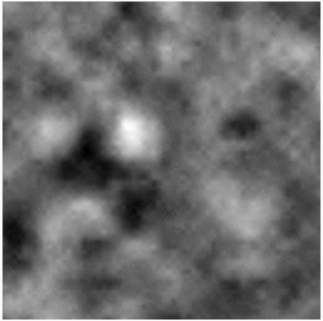
Bayesian Inversion: Samples



$S(x) = \|\nabla x\|_1$

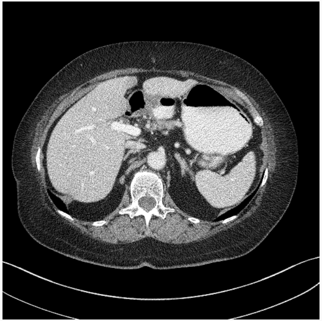
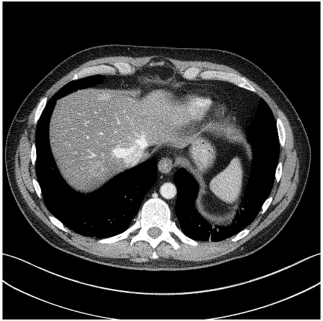


$S(x) = \|x\|_{B^1_{1,1}}$



$S(x) = \|x\|_{B^2_{1,1}}$

Bayesian Inversion: Examples of natural images

Bayesian Inversion: Estimators

- The posterior is enormously high dimensional

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

- Example: 2×2 image, 32 bits/pixel. Dimensionality of the posterior:

$$32^{2\times 2} = 1024 \times 1024$$

  Hence the posterior of 2×2 images is as high dimensional as 1024×1024 images!
- Not even a chance that we could store it.
- All we can hope for is some estimator.

Bayesian Inversion: Conclusions

- Framework for solving inverse problems
- Strong regularizing properties
- Uncertainty quantification
- Basically parameter free
- Classical methods are relatively slow and require closed form prior

Deep Direct Estimation

Estimating uncertanity (and more) using neural networks without $\mathbb{P}(\mathbf{x} \mid \mathbf{y})$

Bayesian Inversion: Hopes and dreams

What would we do if we had $\mathbb{P}(\mathbf{x} \mid \mathbf{y})$?

- Variance

$$\mathbb{E}\left[\left(\mathbf{x} - \mathbb{E}[\mathbf{x} \mid \mathbf{y} = y]\right)^2 \mid \mathbf{y} = y\right]$$

- Covariance

$$\mathbb{E}\left[\left(\mathbf{x}_1 - \mathbb{E}[\mathbf{x}_1 \mid \mathbf{y} = y]\right)\left(\mathbf{x}_2 - \mathbb{E}[\mathbf{x}_2 \mid \mathbf{y} = y]\right) \mid \mathbf{y} = y\right]$$

- Bayesian hypothesis testing

$$\mathbb{P}(\mathbf{x} \in \Omega \mid \mathbf{y} = y) = \mathbb{E}\left[\mathbb{1}_\Omega(\mathbf{x}) \mid \mathbf{y} = y\right]$$

Deep Direct Estimation: The main insight

- The quantities we're looking for have the form

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

  - Mean (reconstruction): $\mathbf{w} = \mathbf{x}$
  - Variance: $\mathbf{w} = \left(\mathbf{x} - \mathbb{E}[\mathbf{x} \mid \mathbf{y} = y]\right)^2$
  - Hypothesis: $\mathbf{w} = \mathbb{1}_{\mathbf{x}_1 > \mathbf{x}_2}$

- Deep neural networks are trained by solving

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

Deep Direct Estimation: Theoretical foundations

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

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

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

Deep Direct Estimation: Theoretical foundations

Let $h\colon Y \to W$ be any measurable function so

$$\mathbb{E}\Big[\big\|h(\mathbf{y}) - \mathbf{w}\big\|_W^2\Big] = \mathbb{E}\Big[\mathbb{E}\big[\big\|h(\mathbf{y}) - \mathbf{w}\big\|_W^2 \mid \mathbf{y}\big]\Big].$$

Next, W is a Hilbert space so we can expand the squared norm:

$$\begin{aligned}\big\|h(\mathbf{y}) - \mathbf{w}\big\|_W^2 &= \big\|h(\mathbf{y}) - \mathbb{E}[\mathbf{w} \mid \mathbf{y}] + \mathbb{E}[\mathbf{w} \mid \mathbf{y}] - \mathbf{w}\big\|_W^2 \\ &= \big\|h(\mathbf{y}) - \mathbb{E}[\mathbf{w} \mid \mathbf{y}]\big\|_W^2 + 2\Big\langle h(\mathbf{y}) - \mathbb{E}[\mathbf{w} \mid \mathbf{y}], \mathbb{E}[\mathbf{w} \mid \mathbf{y}] - \mathbf{w}\Big\rangle_W + \big\|\mathbf{w} - \mathbb{E}[\mathbf{w} \mid \mathbf{y}]\big\|_W^2.\end{aligned}$$

By the law of total expectation and the linearity of the inner product, we get

$$\begin{aligned}&\mathbb{E}\Big[2\Big\langle h(\mathbf{y}) - \mathbb{E}[\mathbf{w} \mid \mathbf{y}], \mathbb{E}[\mathbf{w} \mid \mathbf{y}] - \mathbf{w}\Big\rangle_W \mid \mathbf{y}\Big] \\ &\qquad = 2\Big\langle h(\mathbf{y}) - \mathbb{E}[\mathbf{w} \mid \mathbf{y}], \mathbb{E}[\mathbf{w} \mid \mathbf{y}] - \mathbb{E}[\mathbf{w} \mid \mathbf{y}]\Big\rangle_W = 2\big\langle h(\mathbf{y}) - \mathbb{E}[\mathbf{w} \mid \mathbf{y}], 0\big\rangle_W = 0\end{aligned}$$

and $\big\|\mathbf{w} - \mathbb{E}[\mathbf{w} \mid \mathbf{y}]\big\|_W^2$ is independent of h.

Deep Direct Estimation: Theoretical foundations

Combining all of this gives

$$\operatorname*{arg\,min}_{h\colon Y\to W} \mathbb{E}\left[\left\|h(\mathbf{y})-\mathbf{w}\right\|_W^2\right] = \operatorname*{arg\,min}_{h\colon Y\to W} \mathbb{E}\left[\left\|h(\mathbf{y})-\mathbb{E}[\mathbf{w} \mid \mathbf{y}]\right\|_W^2\right]$$

where $h^*(y) = \mathbb{E}[\mathbf{w} \mid \mathbf{y}]$ is the solution to the right hand side. $\square$

Deep Direct Estimation: Theoretical foundations

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

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

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

Deep Direct Estimation: Computing the mean

- Suppose we want to compute the conditional mean

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

Deep Direct Estimation: Computing the mean

- Suppose we want to compute the conditional mean

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

- We need to find the best measurable function h

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

Deep Direct Estimation: Computing the mean

- Suppose we want to compute the conditional mean

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

- We need to find the best measurable function h

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

- NN are universal approximators, train a neural network $h_\theta : Y \to X$

$$\arg\min_{\theta} \sum_{i=1}^{n} \big\| h_\theta(\mathbf{y}_i) - \mathbf{x}_i \big\|_X^2$$

Deep Direct Estimation: Computing the variance

- Suppose we want to compute the conditional variance

$$\mathbb{E}\left[\left(\mathbf{x} - \mathbb{E}[\mathbf{x} \mid \mathbf{y}]\right)^2 \mid \mathbf{y} = y\right]$$

Deep Direct Estimation: Computing the variance

- Suppose we want to compute the conditional variance

$$\mathbb{E}\Big[\big(\mathbf{x} - \mathbb{E}[\mathbf{x} \mid \mathbf{y}]\big)^2 \mid \mathbf{y} = y\Big]$$

- Note that $\mathbf{w} = \big(\mathbf{x} - \mathbb{E}[\mathbf{x} \mid \mathbf{y}]\big)^2$ so we seek to minimize

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

Deep Direct Estimation: Computing the variance

- Suppose we want to compute the conditional variance

$$\mathbb{E}\left[\left(\mathbf{x} - \mathbb{E}[\mathbf{x} \mid \mathbf{y}]\right)^2 \mid \mathbf{y} = y\right]$$

- Note that $\mathbf{w} = \left(\mathbf{x} - \mathbb{E}[\mathbf{x} \mid \mathbf{y}]\right)^2$ so we seek to minimize

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

- Train neural networks h (mean) and g (variance)

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

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

Deep Direct Estimation: Computing the variance

- Suppose we want to compute the conditional variance

$$\mathbb{E}\Big[\big(\mathbf{x} - \mathbb{E}[\mathbf{x} \mid \mathbf{y}]\big)^2 \mid \mathbf{y} = y\Big]$$

- Note that $\mathbf{w} = \big(\mathbf{x} - \mathbb{E}[\mathbf{x} \mid \mathbf{y}]\big)^2$ so we seek to minimize

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

- Train neural networks h_θ (mean) and g_ϕ (variance)

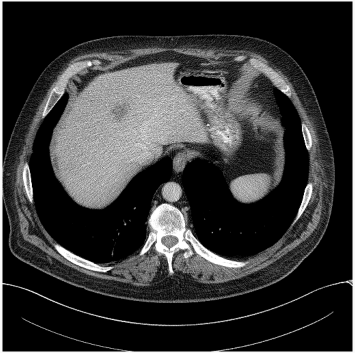
$$\theta^* = \min_\theta \sum_{i=1}^n \big\|h_\theta(\mathbf{y}_i) - \mathbf{x}\big\|_X^2$$

$$\phi^* = \min_\phi \sum_{i=1}^n \big\|g_\phi(\mathbf{y}_i) - \big(\mathbf{x} - h_{\theta^*}(\mathbf{y})\big)^2\big\|_X^2$$

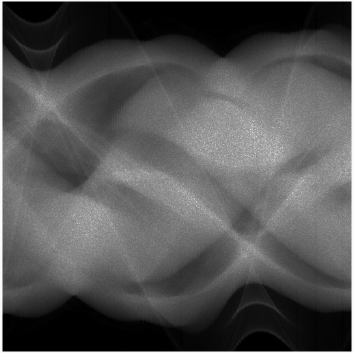
Deep Direct Estimation: Example application

Problem setup:

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



Phantom



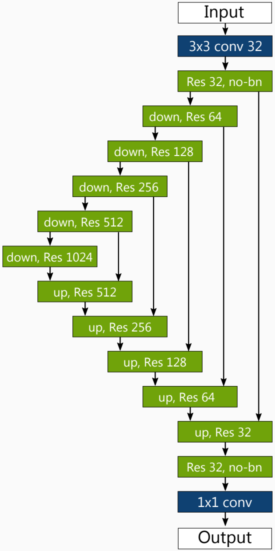
FBP

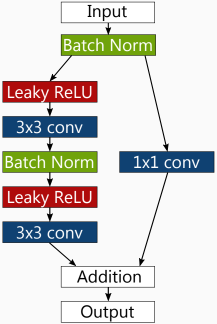
Standard FBP

Deep Direct Estimation: Example application

Solution method:

- Post-processing ☹
- Network architecture $h_\theta = \hat{h}_\theta \circ A^\dagger$
- $\hat{h}_\theta$ is a residual U-Net



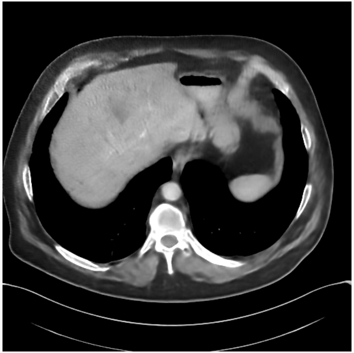


Deep Direct Estimation: Example application

Standard FBP

Deep Direct Estimation: Example application

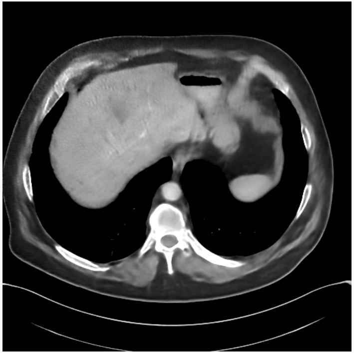
Standard FBP



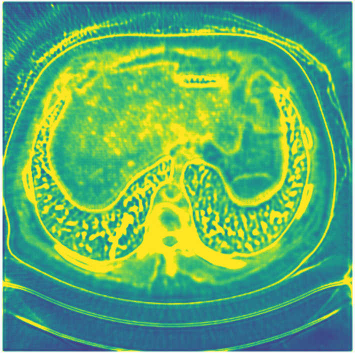
Mean

Deep Direct Estimation: Example application

Standard FBP



Mean



Standard deviation

Deep Direct Estimation: Conclusion

- Most estimators we are interested in can be formulated as

$$\mathbb{E}\Big[\mathbf{w} \mid \mathbf{y} = y\Big]$$

for some $\mathbf{w}$.

- We can characterize this as the solution to

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

- This is *exactly* classical NN training (with a different target)
- We can train networks to find almost any estimator!
- But we need a new network for every estimator...

Deep posterior sampling

Generative models for uncertainty quantification in inverse problems

Bayesian Inversion: Hopes and dreams

What would we do if we had $\mathbb{P}(\mathbf{x} \mid \mathbf{y})$?

- Variance

$$\mathbb{E}\left[\left(\mathbf{x} - \mathbb{E}[\mathbf{x} \mid \mathbf{y} = y]\right)^2 \mid \mathbf{y}\right]$$

- Covariance

$$\mathbb{E}\left[\left(\mathbf{x}_1 - \mathbb{E}[\mathbf{x}_1 \mid \mathbf{y} = y]\right)\left(\mathbf{x}_2 - \mathbb{E}[\mathbf{x}_2 \mid \mathbf{y} = y]\right) \mid \mathbf{y}\right]$$

- Bayesian hypothesis testing

$$\mathbb{P}(\mathbf{x} \in \Omega \mid \mathbf{y} = y) = \mathbb{E}\left[\mathbb{1}_\Omega(\mathbf{x}) \mid \mathbf{y} = y\right]$$

Deep Posterior Sampling: The main insight

- The quantities we're looking for have the form

$$\mathbb{E}\Big[\mathbf{w} \mid \mathbf{y} = y\Big]$$

  - Mean (reconstruction): $\mathbf{w} = \mathbf{x}$
  - Variance: $\mathbf{w} = \big(\mathbf{x} - \mathbb{E}[\mathbf{x} \mid \mathbf{y} = y]\big)^2$
  - Hypothesis: $\mathbf{w} = \mathbb{1}_{\mathbf{x}_1 > \mathbf{x}_2}$
- Law of large numbers: Assume w_i I.I.D. from $\mathbf{w} \mid \mathbf{y} = y$, then a.s.

$$\lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} w_i \to \mathbb{E}\Big[\mathbf{w} \mid \mathbf{y} = y\Big]$$

- All we need is I.I.D. samples!

Generative models in Machine Learning

Input: Training data (x_i) generated by $(\mathbf{x})$.

Goal: Sample from unknown distribution $\mathbb{P}(x)$.

Approaches:

- Variational Auto-Encoders
- Plug and Play Generative Networks
- Pixel Recurrent Models
- Generative Adversarial Networks

Variational Auto-Encoders

- Main idea: train two networks, encoder E and decoder D
- Train to approximate identity

$$\mathbf{x} \approx D(E(\mathbf{x}))$$

  while also enforcing

$$E(\mathbf{x}) \approx \mathbf{z}$$

- Sample from

$$D(\mathbf{z})$$

Plug and Play Generative Networks

- Main idea: Langevin dynamics
- Denoisers approximate gradient

$$\frac{D(x + \epsilon dx) - x}{\epsilon} = \nabla \log \mathbb{P}(x)$$

- Create Markov chain

$$x_{k+1} := x_k + dt \nabla \log \pi(x_k) + \sqrt{2dt}\mathcal{N}(0, 1)$$

Pixel Recurrent Networks

- Main idea: Chain rule
- Expand probability pixel-wise

$$
\begin{aligned}
\mathbb{P}(x) &= \mathbb{P}(x_1, x_2, \dots, x_N) \\
&= \mathbb{P}(x_1)\mathbb{P}(x_2, \dots, x_N \mid x_1) \\
&= \mathbb{P}(x_1)\mathbb{P}(x_2 \mid x_1)\mathbb{P}(x_3, \dots, x_N \mid x_1, x_2) \\
&= \prod_{i=1}^{N} \mathbb{P}(x_i \mid x_{<i})
\end{aligned}
$$

- Each of the $\mathbb{P}(x_i \mid x_{<i})$ are real valued random variables
- Discretize and model explicitly using RNN

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"

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

Wasserstein GAN

Input: Unsupervised data (x_i) generated by $\mathbf{x}$.

Goal: Sample from unknown distribution $\mathbb{P}(x)$.

Approach: Learn how to sample from distribution by solving

$$\min_{\theta} \mathcal{W}\big(\mathrm{G}_{\theta}, \mathbb{P}(\mathbf{x})\big)$$

Wasserstein GAN

Input: Unsupervised data (x_i) generated by $\mathbf{x}$.

Goal: Sample from unknown distribution $\mathbb{P}(x)$.

Approach: Learn how to sample from distribution by solving

$$\min_{\theta} \mathcal{W}\big(\mathrm{G}_\theta, \mathbb{P}(\mathbf{x})\big)$$

- G_θ is a probability distribution on model parameters in X.
- $\mathcal{W}$ is the Wasserstein 1-distance, measures how close G_θ is to the distribution.

Wasserstein GAN

Input: Unsupervised data (x_i) generated by $\mathbf{x}$.

Goal: Sample from unknown distribution $\mathbb{P}(x)$.

Approach: Learn how to sample from distribution by solving

$$\min_\theta \mathcal{W}\big(\mathrm{G}_\theta, \mathbb{P}(\mathbf{x})\big)$$

Unfeasible: Not possible to evaluate $\mathcal{W}$ ($\mathbb{P}(\mathbf{x})$ unknown).

Wasserstein GAN

Input: Unsupervised data (x_i) generated by $\mathbf{x}$.

Goal: Sample from unknown distribution $\mathbb{P}(x)$.

Approach: Learn how to sample from distribution by solving

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

Unfeasible: Not possible to evaluate $\mathcal{W}$ ($\mathbb{P}(\mathbf{x})$ unknown).

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

Wasserstein GAN

Input: Unsupervised data (x_i) generated by $\mathbf{x}$.

Goal: Sample from unknown distribution $\mathbb{P}(x)$.

Approach: Learn how to sample from distribution by solving

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

Unfeasible: Maximization over *all* Lipschitz operators

Wasserstein GAN

Input: Unsupervised data (x_i) generated by $\mathbf{x}$.

Goal: Sample from unknown distribution $\mathbb{P}(x)$.

Approach: Learn how to sample from distribution by solving

$$\min_{\theta} \left\{ \max_{\phi} \mathbb{E}\left[\mathrm{D}_{\phi}(\mathbf{x}) - \mathrm{D}_{\phi}(\mathrm{G}_{\theta}) \right] \right\}.$$

Unfeasible: Maximization over *all* Lipschitz operators

$\implies$ Let discriminator be a NN.

Wasserstein GAN

Input: Unsupervised data (x_i) generated by $\mathbf{x}$.

Goal: Sample from unknown distribution $\mathbb{P}(x)$.

Approach: Learn how to sample from distribution by solving

$$\min_{\theta} \left\{ \max_{\phi} \mathbb{E}\left[\mathrm{D}_{\phi}(\mathbf{x}) - \mathrm{D}_{\phi}(\mathrm{G}_{\theta}) \right] \right\}.$$

Unfeasible: How is G_θ random?

Wasserstein GAN

Input: Unsupervised data (x_i) generated by $\mathbf{x}$.

Goal: Sample from unknown distribution $\mathbb{P}(x)$.

Approach: Learn how to sample from distribution by solving

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

Unfeasible: How is G_{θ} random?

$\implies$ Write as deterministic function of random input

Wasserstein GAN

Input: Unsupervised data (x_i) generated by $\mathbf{x}$.

Goal: Sample from unknown distribution $\mathbb{P}(x)$.

Approach: Learn how to sample from distribution by solving

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

Unfeasible: Expectation over samples

Wasserstein GAN

Input: Unsupervised data (x_i) generated by $\mathbf{x}$.

Goal: Sample from unknown distribution $\mathbb{P}(x)$.

Approach: Learn how to sample from distribution by solving

$$\min_{\theta} \left\{ \max_{\phi} \left[\frac{1}{N} \sum_{i=1}^{N} \mathrm{D}_{\phi}(x_i) - \mathbb{E}_{\mathbf{z}} \, \mathrm{D}_{\phi}(\mathrm{G}_{\theta}(\mathbf{z})) \right] \right\}.$$

Unfeasible: Expectation over samples

$\implies$ Use empirical distribution

Wasserstein GAN

Input: Unsupervised data (x_i) generated by $\mathbf{x}$.

Goal: Sample from unknown distribution $\mathbb{P}(x)$.

Approach: Learn how to sample from distribution by solving

$$\min_{\theta}\left\{\max_{\phi}\left[\frac{1}{N}\sum_{i=1}^{N}\mathrm{D}_{\phi}(x_i) - \mathbb{E}_{\mathbf{z}}\,\mathrm{D}_{\phi}(\mathrm{G}_{\theta}(\mathbf{z}))\right]\right\}.$$

Approximation to Wasserstein distance useful for deep learning

Conditional Wasserstein GAN

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

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

Approach: Learn how to sample from posterior by solving

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

Conditional Wasserstein GAN

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

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

Approach: Learn how to sample from posterior by solving

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

Condition on data **y**, else same steps above (with some technical additions)

Conditional Wasserstein GAN

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

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

Approach: Learn how to sample from posterior by solving

$$\min_{\theta} \left\{ \max_{\phi} \frac{1}{N} \sum_{i=1}^{N} \left[\mathrm{D}_{\phi}(x_i, y_i) - \mathbb{E}_{\mathbf{z}} \left[\mathrm{D}_{\phi}(\mathrm{G}_{\theta}(\mathbf{z}, y_i), y_i) \right] \right] \right\}.$$

Condition on data $\mathbf{y}$, else same steps above (with some technical additions)

Conditional Wasserstein GAN

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

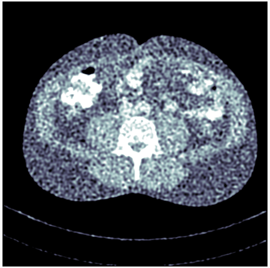
Goal: Sample from unknown posterior $\mathbb{P}(x \mid y)$.

Approach: Learn how to sample from posterior by solving

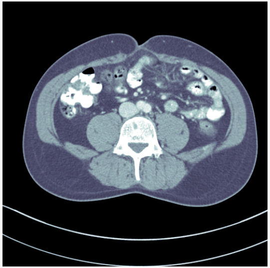
$$\min_{\theta} \left\{ \max_{\phi} \frac{1}{N} \sum_{i=1}^{N} \left[\mathrm{D}_{\phi}(x_i, y_i) - \mathbb{E}_{\mathbf{z}} \left[\mathrm{D}_{\phi}(\mathrm{G}_{\theta}(\mathbf{z}, y_i), y_i) \right] \right] \right\}.$$

Formulation useful for deep learning

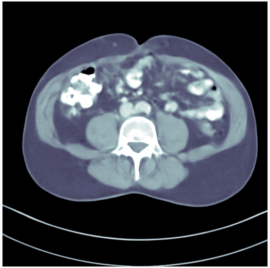
One of the images is the ground truth (phantom), can you figure out which one?



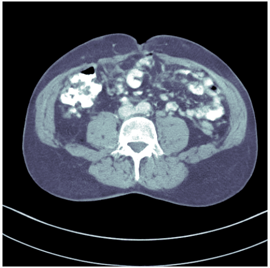
1



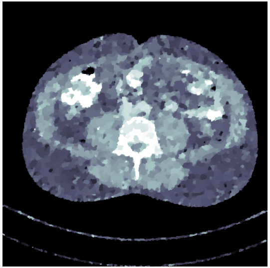
2



3

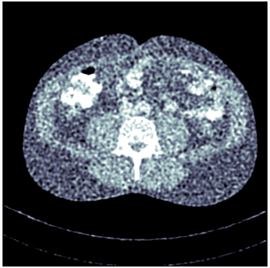


4

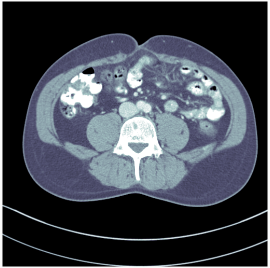


5

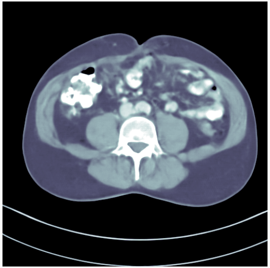
One of the images is the ground truth (phantom), can you figure out which one?



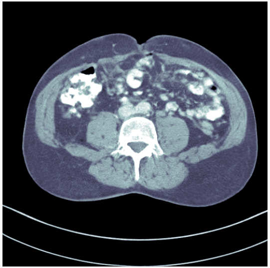
FBP



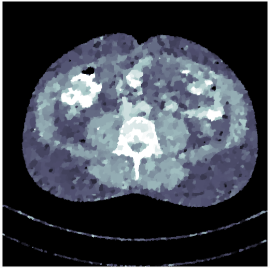
2



Conditional mean

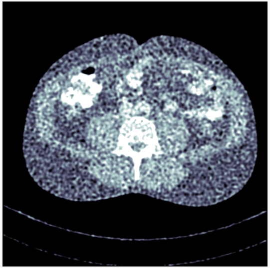


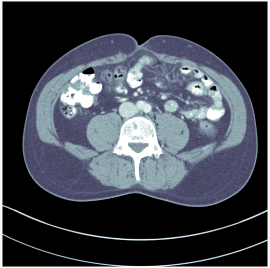
4

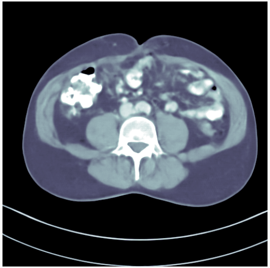


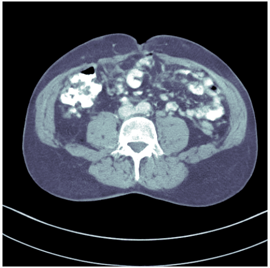
Total variation

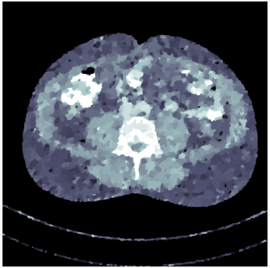
One of the images is the ground truth (phantom), can you figure out which one?

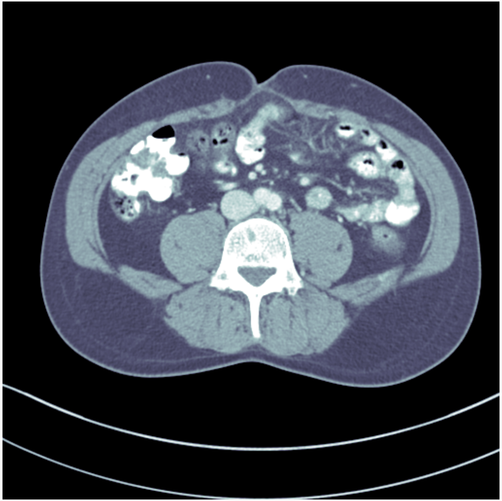

FBP


Phantom

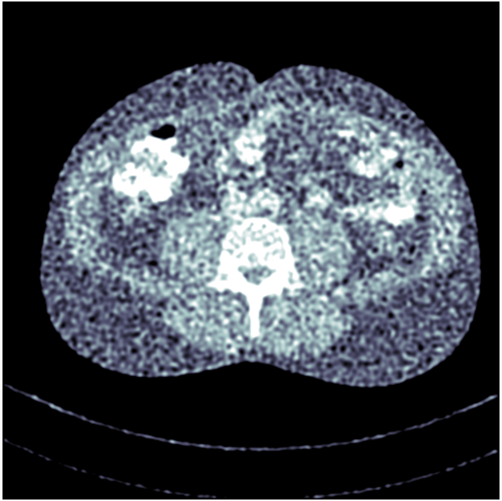

Conditional mean


Deep posterior sample

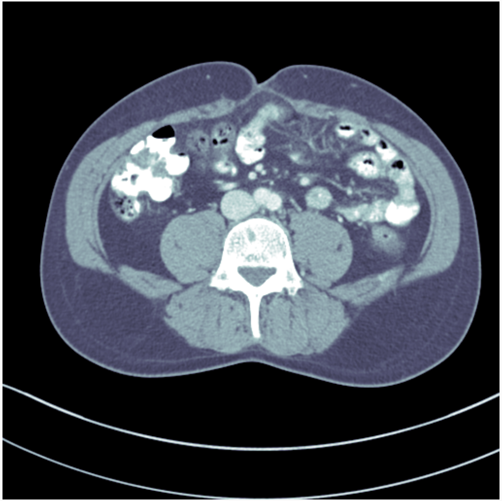

Total variation



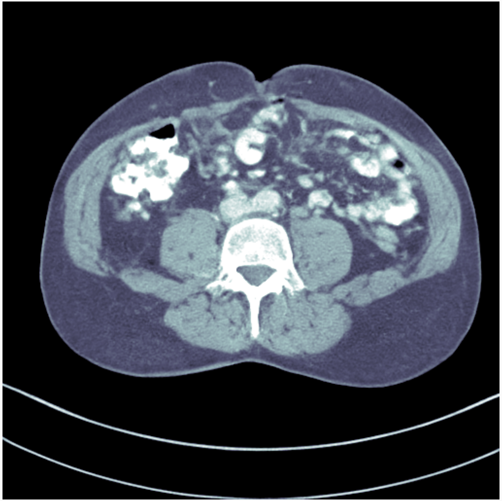
Phantom



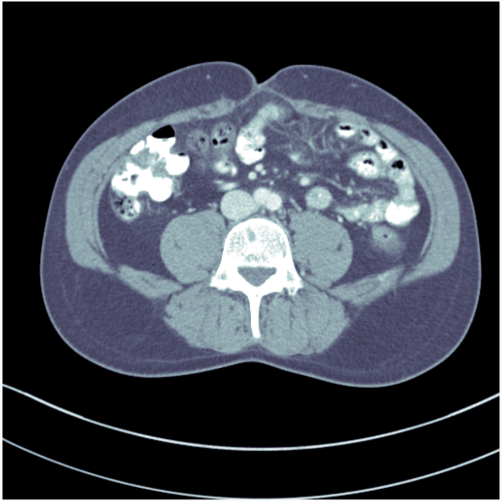
FBP reconstruction



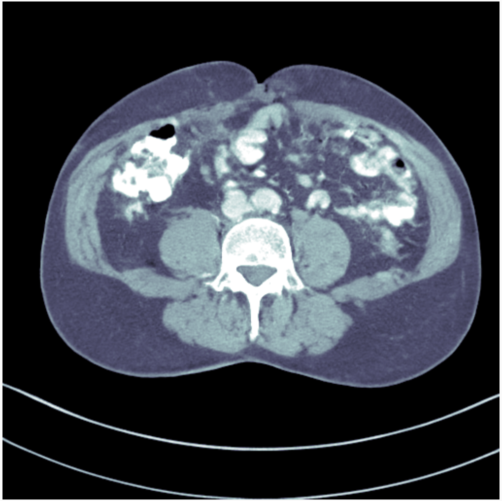
Phantom



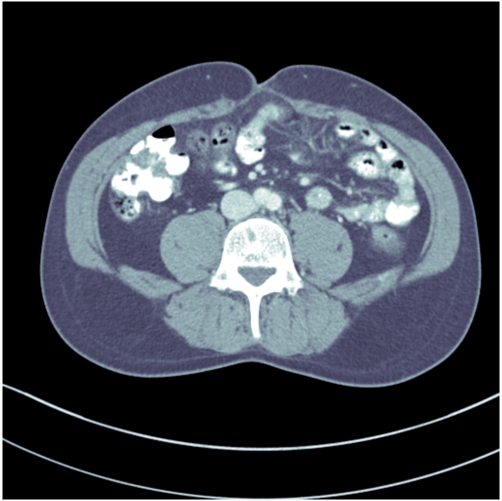
Posterior sample 1



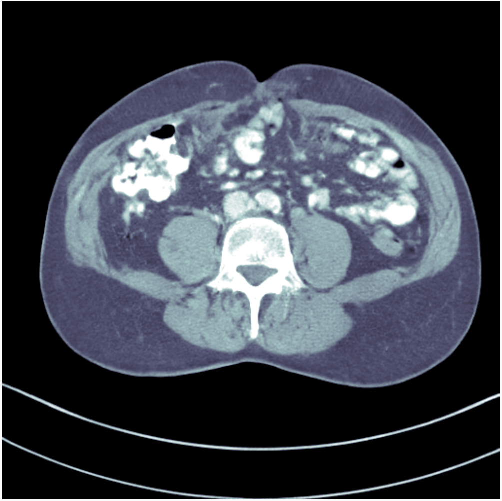
Phantom



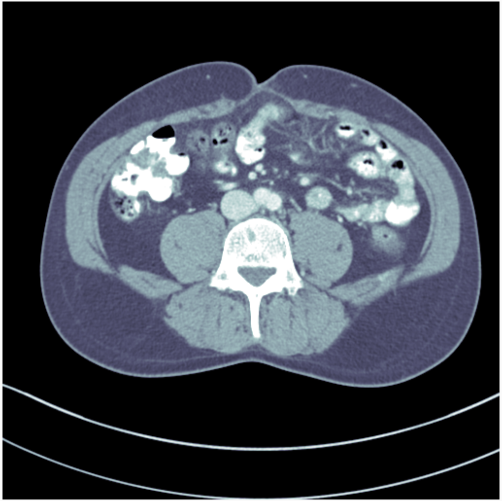
Posterior sample 2



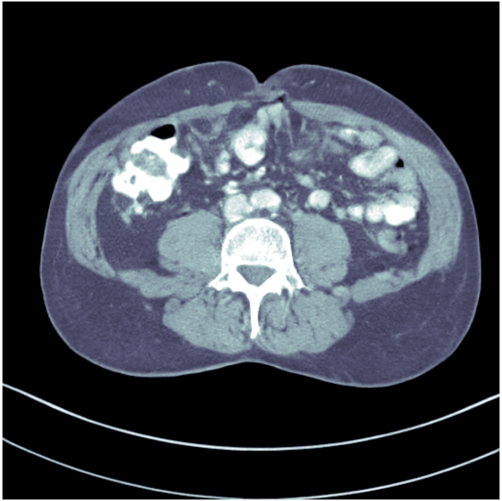
Phantom



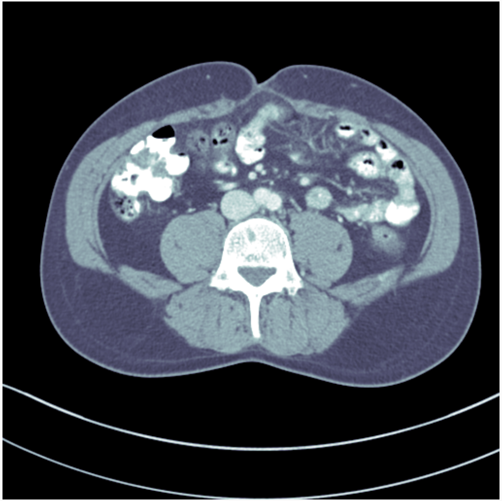
Posterior sample 3



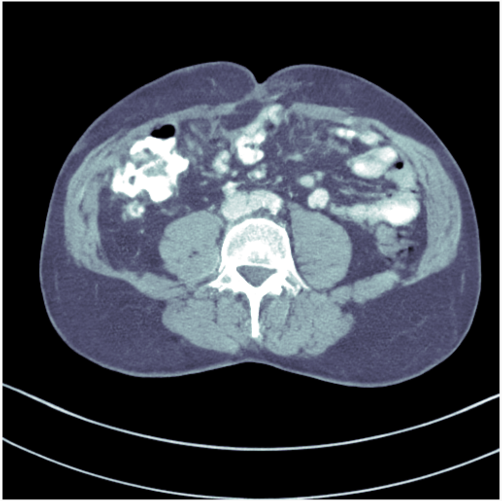
Phantom



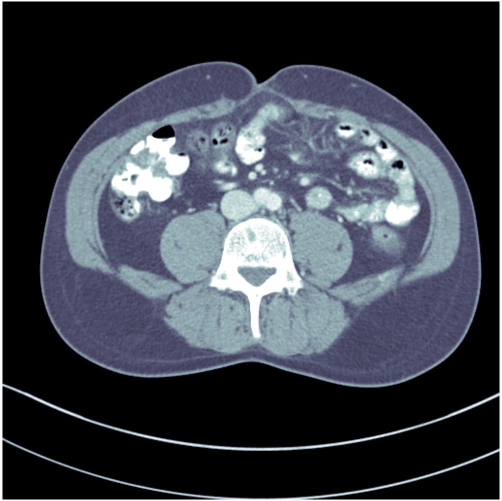
Posterior sample 4



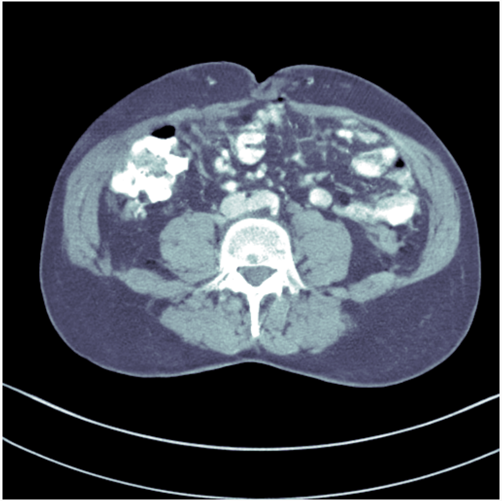
Phantom



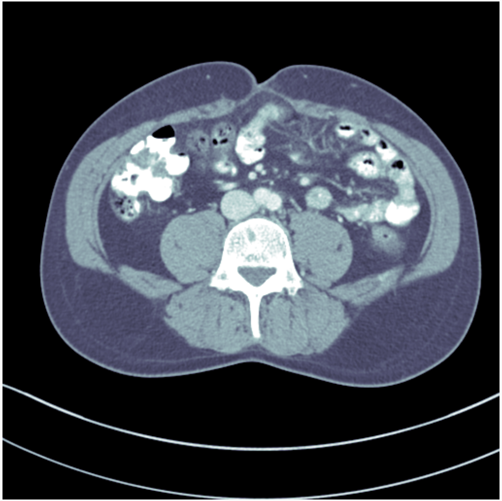
Posterior sample 5



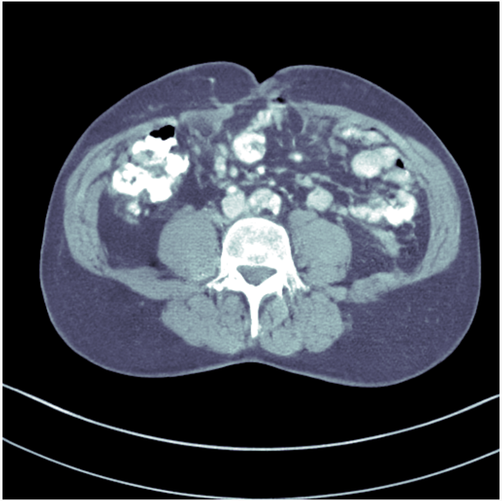
Phantom



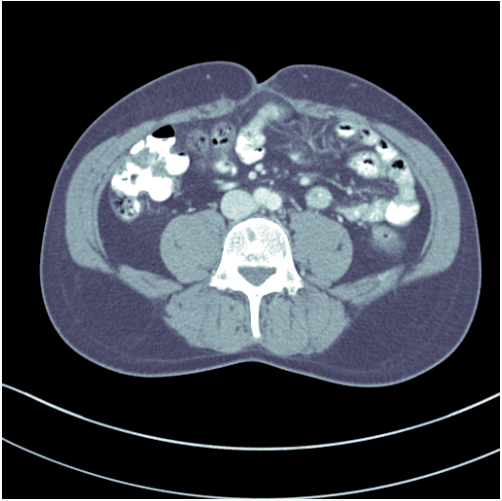
Posterior sample 6



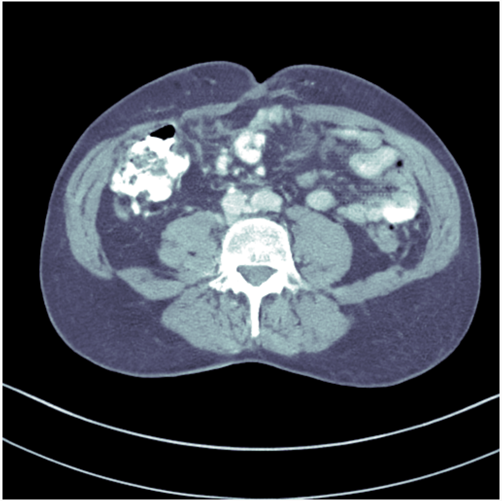
Phantom



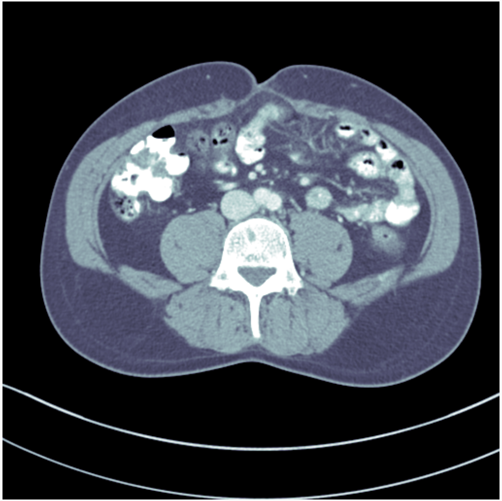
Posterior sample 7



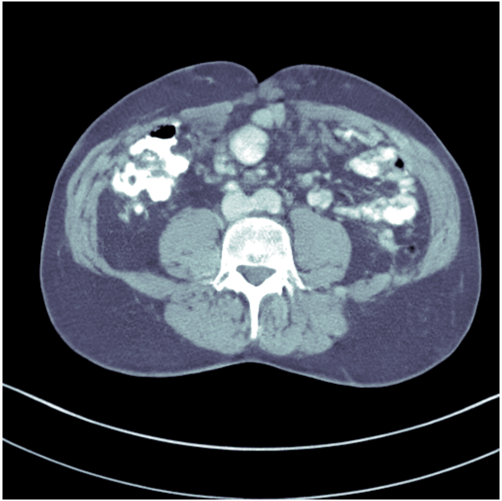
Phantom



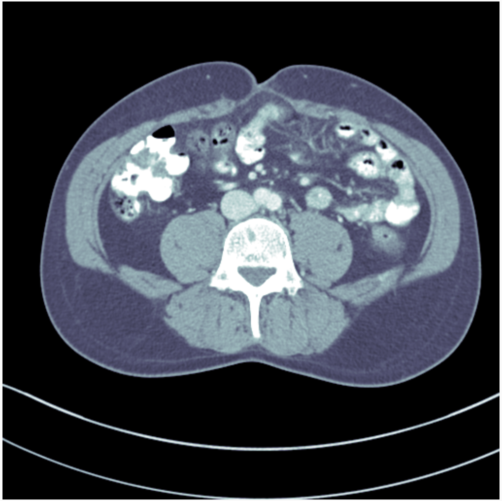
Posterior sample 8



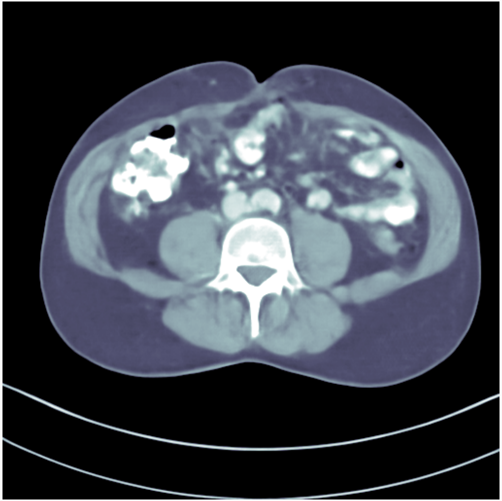
Phantom



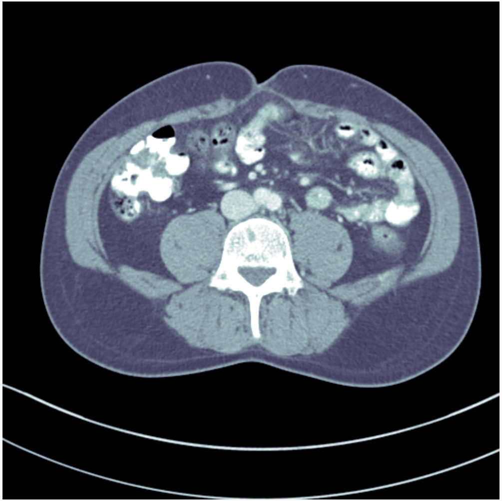
Posterior sample 9



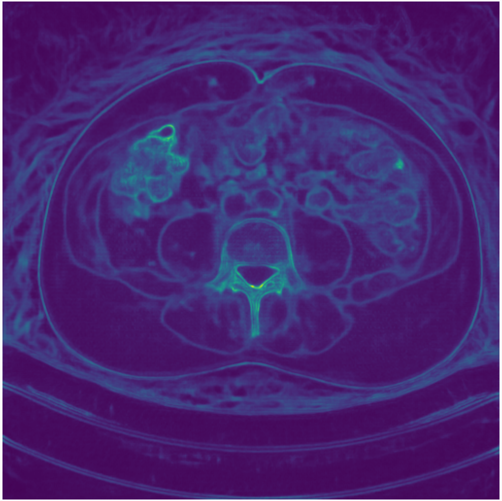
Phantom



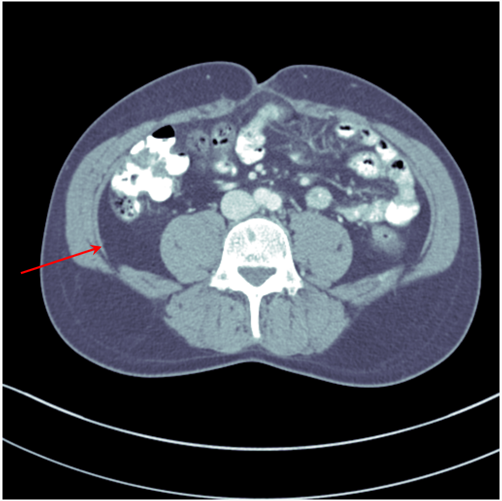
Conditional mean (1000 samples)



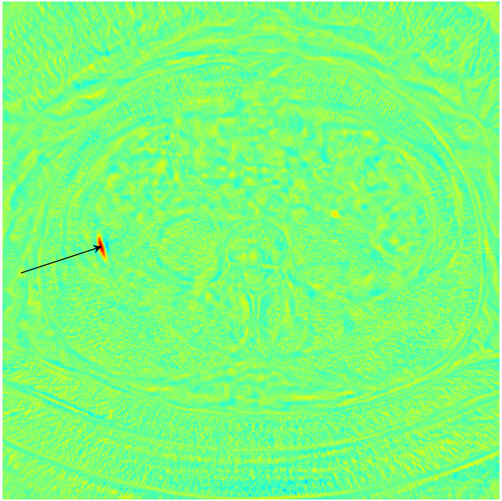
Phantom



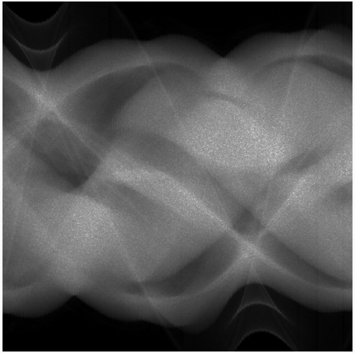
Standard deviation



Phantom



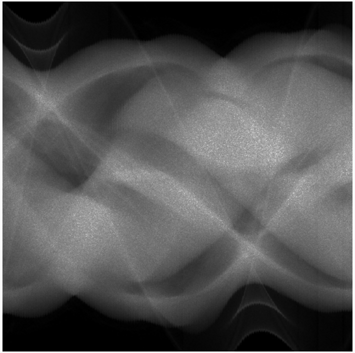
Correlation w.r.t. single point



Data

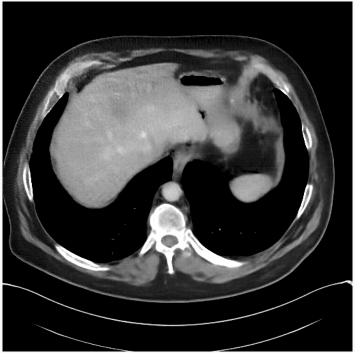
FBP

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



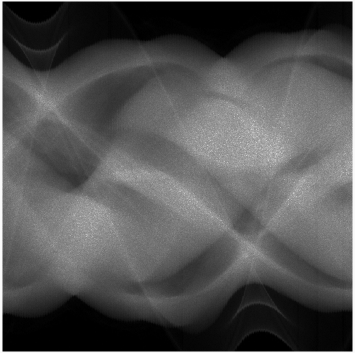
Data

FBP



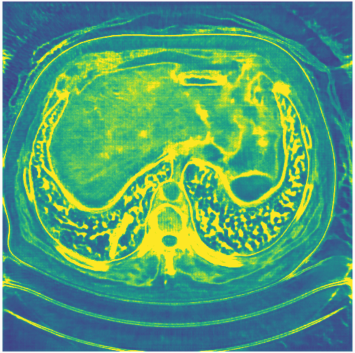
Posterior mean

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



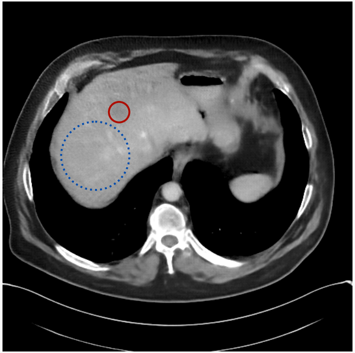
Data

FBP



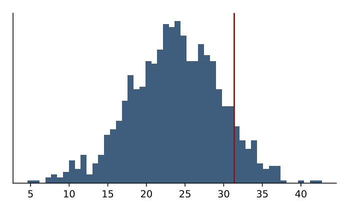
Standard deviation

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

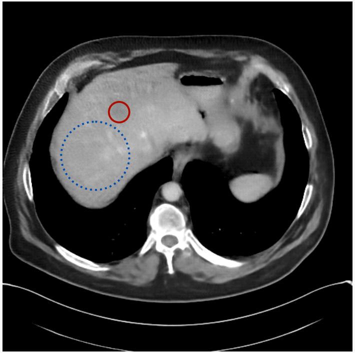


Posterior mean with ROI

- 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%

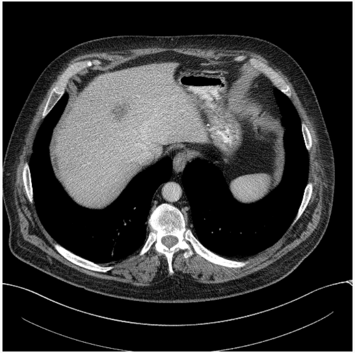


Histogram of $\triangle$

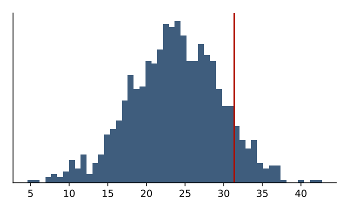


Posterior mean with ROI

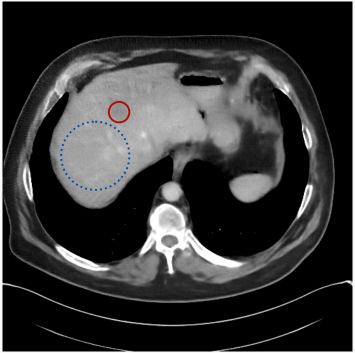
- Case: Patient with suspected metastasis to the liver.
- Data: Clinical helical 3D CT data, 2% of a normal dose (ultra low-dose).
- Liver lesion: $\triangle$ = difference in average contrast between ROI and liver.
- Hypothesis test: Based on 1 000 samples, the ROI contains a lesion at 95%



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%

Deep Posterior Sampling: Conclusion

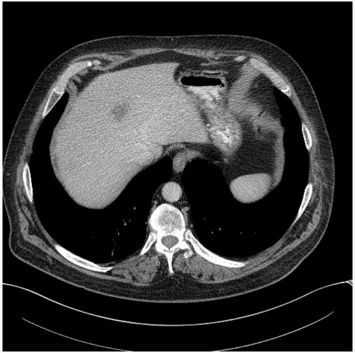
- Most estimators we are interested in can be formulated as

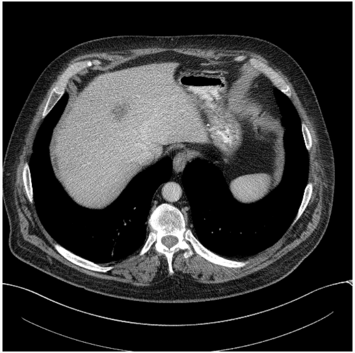
$$\mathbb{E}\left[\mathbf{w} \mid \mathbf{y} = y\right]$$

- We compute them by sampling
- Train a neural network to generate samples (here GAN)
- Can compute any estimator on the fly

Conclusion

- Bayesian Inversion is an extremely powerful framework
- Historical problems with computational feasibility
- Deep Learning methods allow us to compute any estimator quickly and with the "true" prior
- If we know the estimator a-priori: Deep Direct Estimation
- On the fly: Posterior Sampling





Thank you for your attention!