Solving Inverse Problems with Scientific Machine Learning (SciML)

SeRC 12:th Annual Meeting

Ozan Öktem

24 May, 2021

Department of Mathematics
KTH - Kungliga Tekniska Högskolan

SciML and inverse problems

Inverse problems: Recover $x \in X$ given $y \approx A(x)$ and $A \colon X \to Y$

Challenges: Ill-posed, large-scale

SciML and inverse problems

Inverse problems: Recover $x \in X$ given $y \approx A(x)$ and $A \colon X \to Y$

Scientific machine learning (SciML) = Scientific computing + Machine learning

- Domain adapted networks: Incorporate handcrafted models/criteria/constraints.

SciML and inverse problems

Inverse problems: Recover $x \in X$ given $y \approx A(x)$ and $A\colon X \to Y$

Scientific machine learning (SciML) = Scientific computing + Machine learning

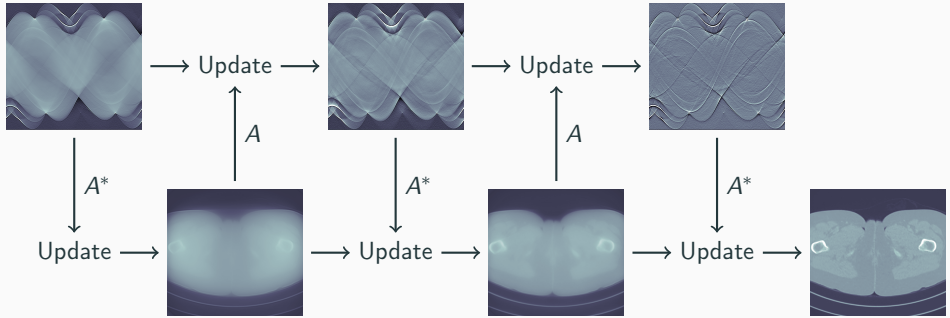
- Domain adapted networks: Incorporate handcrafted models/criteria/constraints.
- Inverse problems: DNN $\approx A^{-1}$, account for explicit $A\colon X \to Y$ and $\partial A^*\colon Y \to L(Y, X)$

SciML and inverse problems

Inverse problems: Recover $x \in X$ given $y \approx A(x)$ and $A\colon X \to Y$

Scientific machine learning (SciML) = Scientific computing + Machine learning

- Domain adapted networks: Incorporate handcrafted models/criteria/constraints.
- Inverse problems: DNN $\approx A^{-1}$, account for explicit $A\colon X \to Y$ and $\partial A^*\colon Y \to L(Y, X)$
- Unrolling: Neural networks for approximating operators defined by iterative schemes.

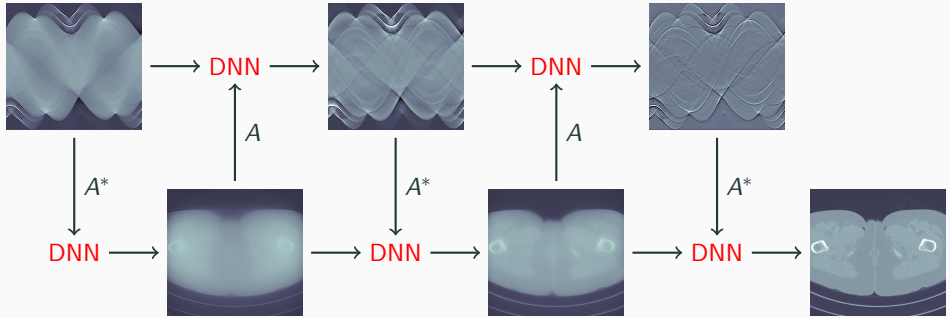


SciML and inverse problems

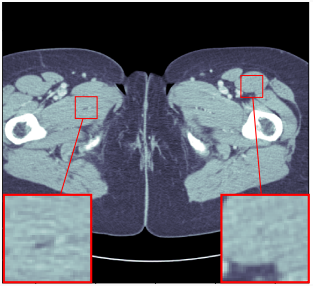
Inverse problems: Recover $x \in X$ given $y \approx A(x)$ and $A\colon X \to Y$

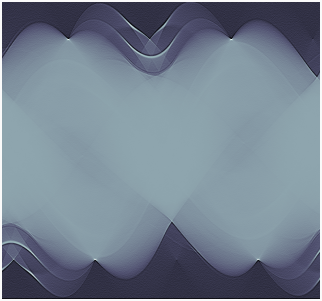
Scientific machine learning (SciML) = Scientific computing + Machine learning

- Domain adapted networks: Incorporate handcrafted models/criteria/constraints.
- Inverse problems: DNN $\approx A^{-1}$, account for explicit $A\colon X \to Y$ and $\partial A^*\colon Y \to L(Y, X)$
- Unrolling: Neural networks for approximating operators defined by iterative schemes.

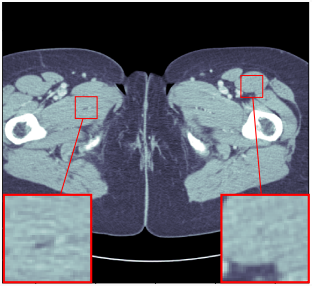


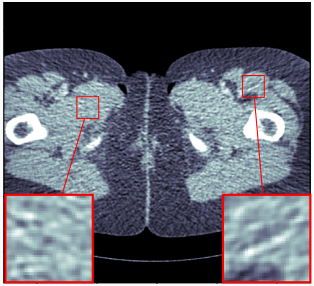
Posterior mean: 2D tomography



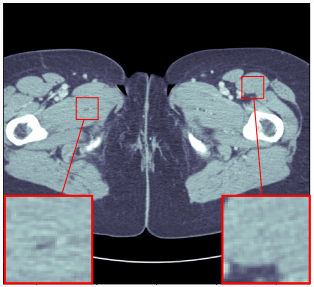


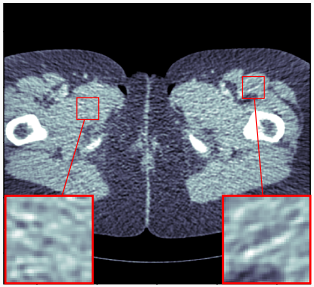
Posterior mean: 2D tomography

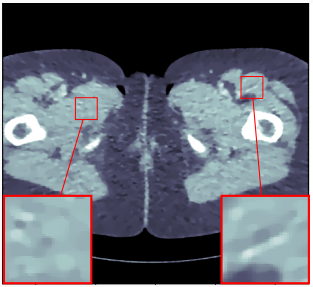




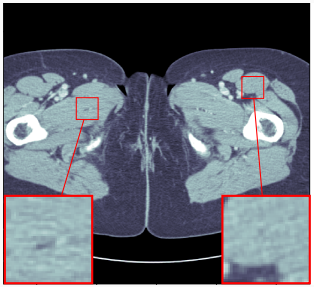
Posterior mean: 2D tomography

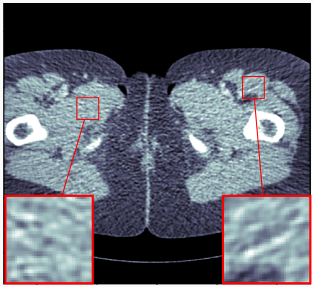


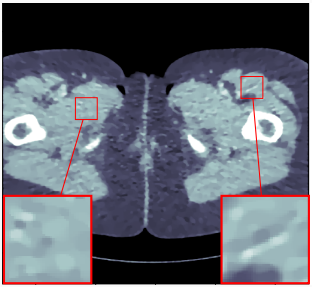


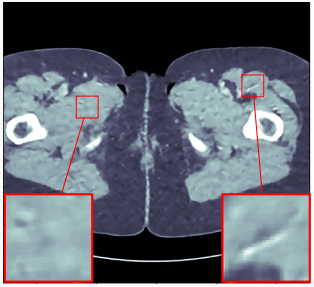


Posterior mean: 2D tomography

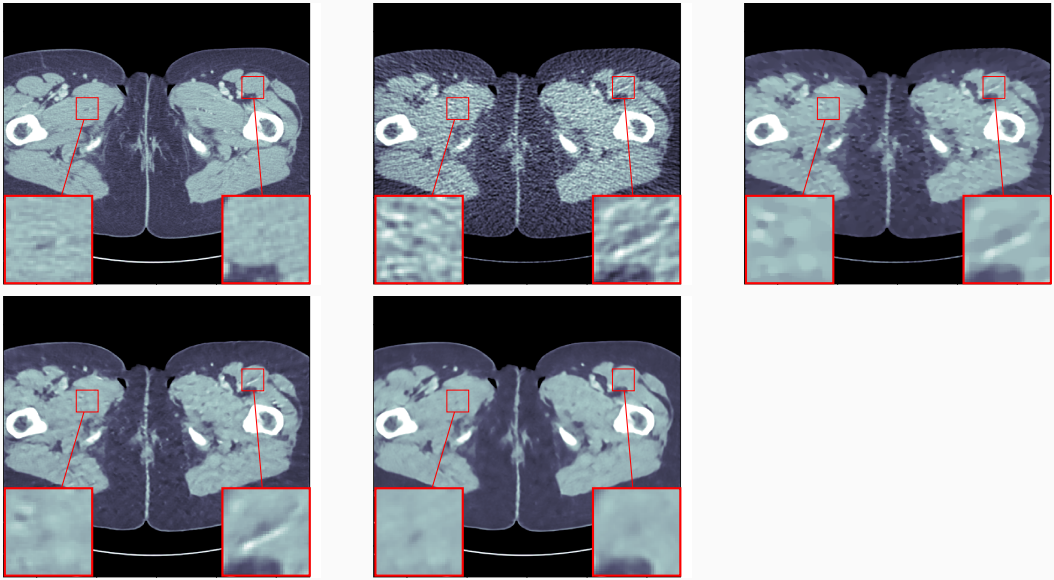




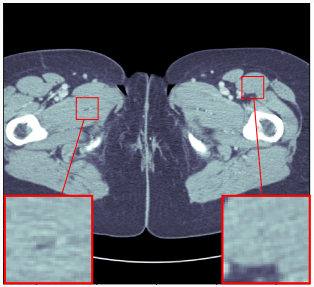


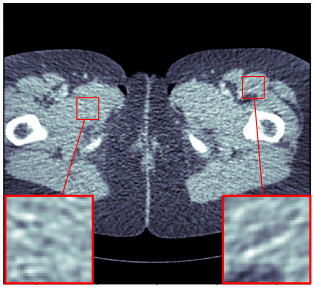


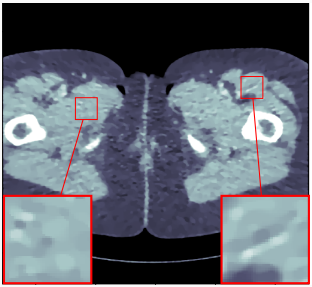
Posterior mean: 2D tomography

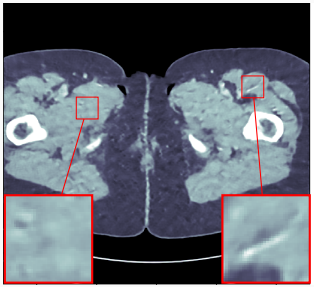


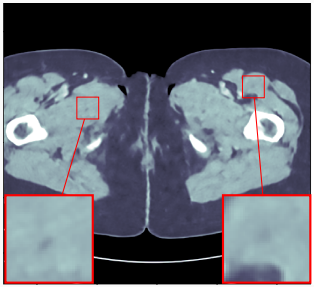
Posterior mean: 2D tomography





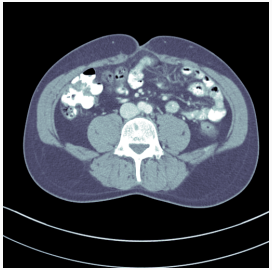




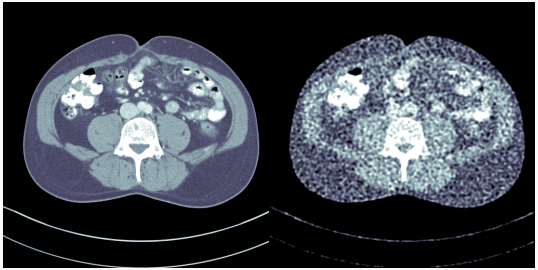


- $\mathbb{y} = A(\mathbb{x}) + \mathbb{e}$
- $y =$ sample from $(\mathbb{y} \mid \mathbb{x} = x^*)$
  x^* unknown
- Reco.: $R_{\widehat{\theta}}(y) \approx \mathbb{E}[\mathbb{x} \mid \mathbb{y} = y]$
  $$\widehat{\theta} \in \operatorname*{argmin}_{\theta} \mathbb{E}_{(\mathbb{x},\mathbb{y})} \left[\left\| R_\theta(\mathbb{y}) - \mathbb{x} \right\|_2^2 \right]$$
- $R_\theta =$ domain adapted DNN
- Training data: Samples of $(\mathbb{x}, \mathbb{y})$

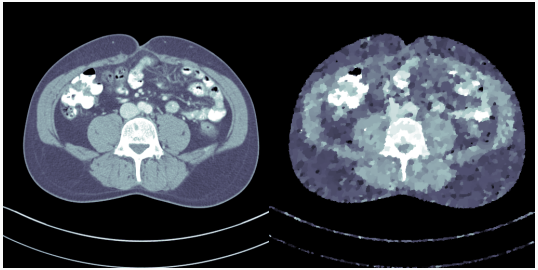
Posterior sampling: 2D tomography



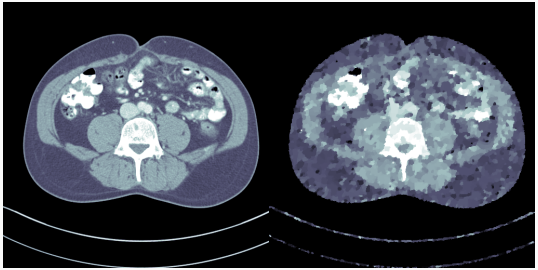
Posterior sampling: 2D tomography

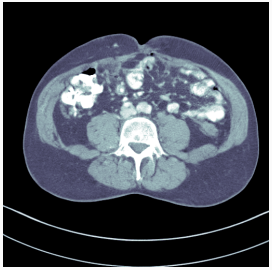


Posterior sampling: 2D tomography



Posterior sampling: 2D tomography





Generative adversarial networks (GANs): Mini-max game between two DNNs (generator/discriminator)

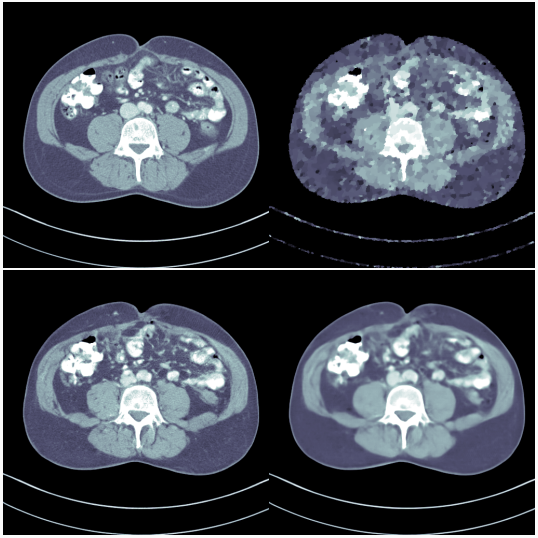
- Conditional GAN: Conditional variant of a GAN, can be used to sample from posterior distribution in an inverse problem.
- WGAN: Minimizes Wasserstein loss

Training: $\widehat{\theta}$ solves conditional WGAN problem:

$$\operatorname*{argmin}_{\theta}\left\{\max_{\phi}\left\{\mathbb{E}_{(\mathbb{x},\mathbb{y})}\left[D_{\phi}(\mathbb{x},\mathbb{y})-\mathbb{E}_{\mathbb{z}}\left[D_{\phi}\left(G_{\theta}(\mathbb{z},\mathbb{y}),\mathbb{y}\right)\right]\right]\right\}\right\}$$

- G_θ = DNN generator
- D_ϕ = DNN discriminator
- Training data: Samples of $(\mathbb{x},\mathbb{y})$
- Trained generator: $G_{\widehat{\theta}}(\mathbb{z},y)\approx \mathsf{P}(\mathbb{x}\mid \mathbb{y}=y)$

Posterior sampling: 2D tomography



Generative adversarial networks (GANs): Mini-max game between two DNNs (generator/discriminator)

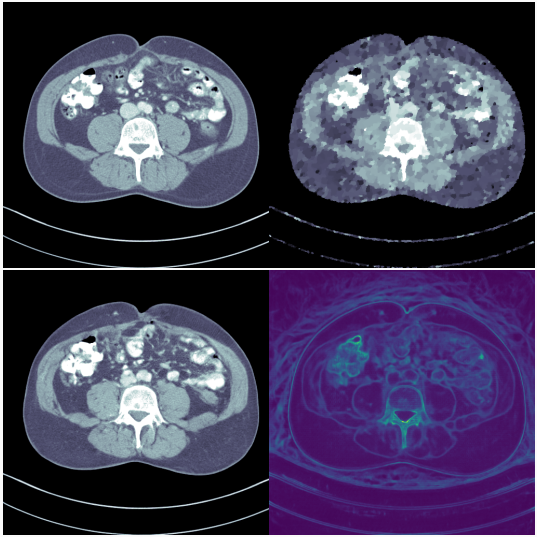
- Conditional GAN: Conditional variant of a GAN, can be used to sample from posterior distribution in an inverse problem.
- WGAN: Minimizes Wasserstein loss

Training: $\widehat{\theta}$ solves conditional WGAN problem:

$$\operatorname*{argmin}_{\theta}\left\{\max_{\phi}\left\{\mathbb{E}_{(\mathbb{x},\mathbb{y})}\left[D_{\phi}(\mathbb{x},\mathbb{y})-\mathbb{E}_{\mathbb{z}}\left[D_{\phi}\left(G_{\theta}(\mathbb{z},\mathbb{y}),\mathbb{y}\right)\right]\right]\right\}\right\}$$

- G_θ = DNN generator
- D_ϕ = DNN discriminator
- Training data: Samples of $(\mathbb{x},\mathbb{y})$
- Trained generator: $G_{\widehat{\theta}}(\mathbb{z},y) \approx \mathsf{P}(\mathbb{x} \mid \mathbb{y}=y)$

Posterior sampling: 2D tomography



Generative adversarial networks (GANs): Mini-max game between two DNNs (generator/discriminator)

- Conditional GAN: Conditional variant of a GAN, can be used to sample from posterior distribution in an inverse problem.
- WGAN: Minimizes Wasserstein loss

Training: $\widehat{\theta}$ solves conditional WGAN problem:

$$\operatorname*{argmin}_{\theta}\left\{\max_{\phi}\left\{\mathbb{E}_{(\mathbb{x},\mathbb{y})}\left[D_{\phi}(\mathbb{x},\mathbb{y})-\mathbb{E}_{\mathbb{z}}\left[D_{\phi}\left(G_{\theta}(\mathbb{z},\mathbb{y}),\mathbb{y}\right)\right]\right]\right\}\right\}$$

- G_θ = DNN generator
- D_ϕ = DNN discriminator
- Training data: Samples of $(\mathbb{x},\mathbb{y})$
- Trained generator: $G_{\widehat{\theta}}(\mathbb{z},y)\approx \mathrm{P}(\mathbb{x}\mid \mathbb{y}=y)$

More details in tomorrow's Digital Futures 'Fly High & Dive Deep' seminar

- Title: Scientific Machine Learning: An Overview with Applications to Inverse Problems
- Date and time: 15:00 – 16:00 CEST (UTC +2)
- Zoom: https://kth-se.zoom.us/j/69560887455
  Meeting ID: 695 6088 7455
  Password: 755440