Deep Learning and Mathematical Modeling: Taking the Best out of Both Worlds

Gitta Kutyniok

(Technische Universität Berlin)

Big Data Science in Astroparticle Research – Workshop
RWTH Aachen, February 18–20, 2019

The 21st Century

Various technological advances in the 21st century are only possible through *integrated mathematical modeling, simulation, and optimization*.

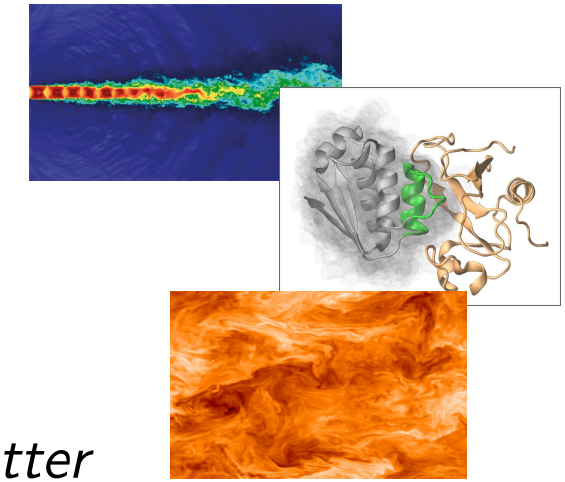
The 21st Century

Various technological advances in the 21st century are only possible through *integrated mathematical modeling, simulation, and optimization*.



Further Examples:

- Turbines
  $\rightsquigarrow$ *Adjoint based jet-noise minimization*
- Atomistic molecular dynamics
  $\rightsquigarrow$ *Simulations with ultralong timescales*
- Star formation
  $\rightsquigarrow$ *Understanding of turbulent accretion of matter*

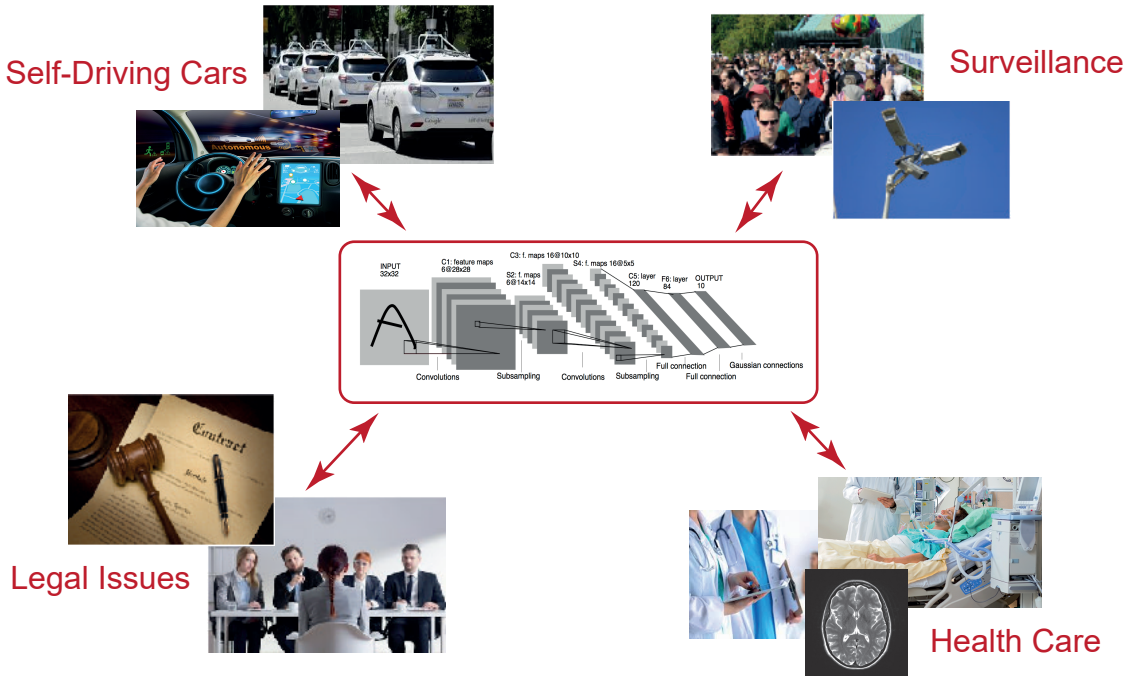


There is a pressing need to go beyond pure modeling, simulation, and optimization approaches!

Data-Based Approaches

Diverse Machine Learning Approaches:

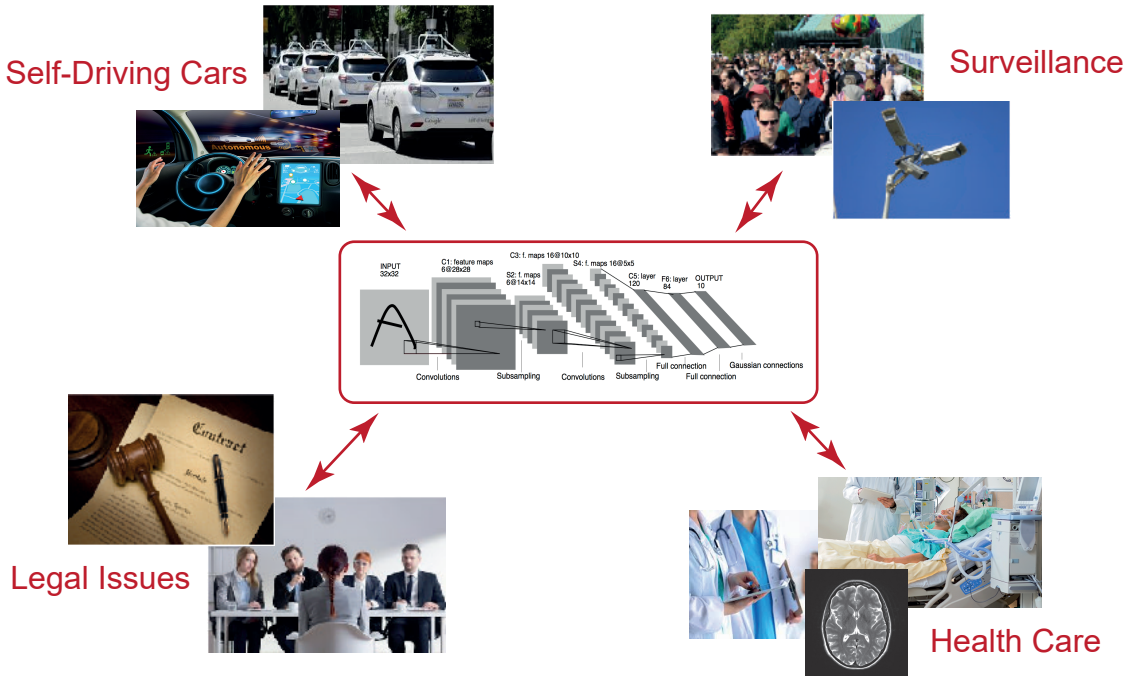
- Principal components analysis
- Support vector machines
- *Deep neural networks*
- ...



Data-Based Approaches

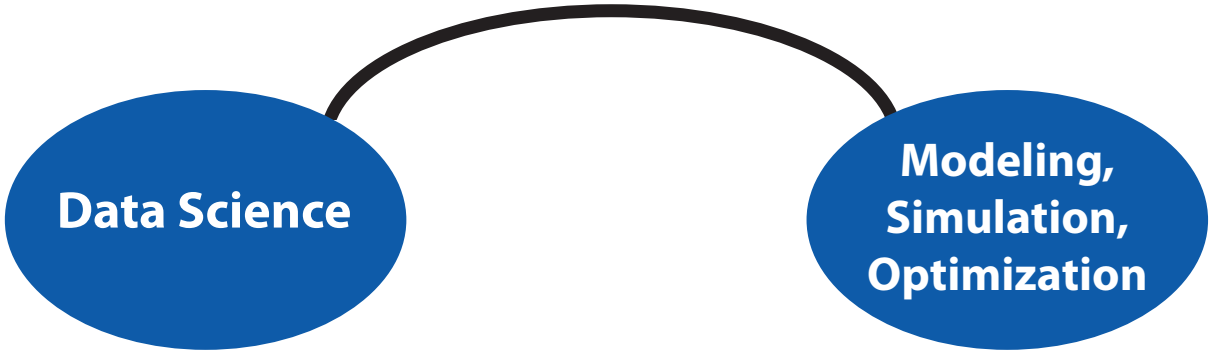
Diverse Machine Learning Approaches:

- Principal components analysis
- Support vector machines
- *Deep neural networks*
- ...



Very few theoretical results explaining their success!

Bridging both Worlds



Current Situation:

- Data Science
  - Novel powerful, pure data-based methodologies
  - Deep theoretical understanding often missing
- Modeling, Simulation, Optimization
  - Traditional (differential-equation based) methodologies very well understood
  - Complexity of current physical or engineering systems too massive

From Data-Driven to Model-Based Approaches

Problems, Viewpoints and Solution Strategies:

- Pure data-driven approaches.
  Detect structural components in data sets!
- Machine learning with physical constraints.
  Insert physical information in machine learning algorithm!
- Parametric differential equations.
  Learn parameters from given data sets!
- Data assimilation.
  Combine sparse data with physical model to generate a general model!
- Data analysis on simulation data.
  Study simulation generated data in search of underlying laws!

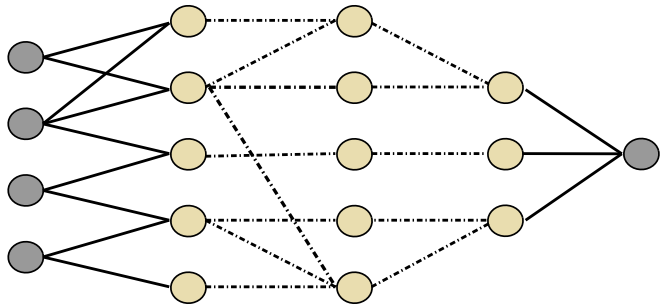
Optimal balancing of
data-adaptiveness and differential equation-based modeling!

Delving Deeper into Deep Neural Networks

Neural Networks from a Mathematical Perspective

Definition:

Assume the following notions:



- $d \in \mathbb{N}$: Dimension of input layer.
- L: Number of layers.
- N: Number of neurons.
- $\sigma : \mathbb{R} \to \mathbb{R}$: (Non-linear) function called *rectifier*.
- $W_\ell : \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell}$, $\ell = 1, \dots, L$: Affine linear maps $(x \mapsto Ax + b)$

Then $\Phi : \mathbb{R}^d \to \mathbb{R}^{N_L}$ given by

$$\Phi(x) = W_L \sigma(W_{L-1} \sigma(\dots \sigma(W_1(x))), \quad x \in \mathbb{R}^d,$$

is called a *(deep) neural network (DNN)*. A DNN with only few non-zero weights is called *sparsely connected*.

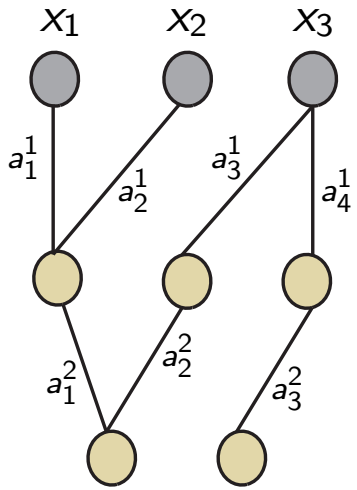
Looking closer...

Remark: The affine linear map W_ℓ is defined by a matrix $A_\ell \in \mathbb{R}^{N_{\ell-1} \times N_\ell}$ and an affine part $b_\ell \in \mathbb{R}^{N_\ell}$ via

$$W_\ell(x) = A_\ell x + b_\ell.$$

$$A_1 = \begin{pmatrix} a_1^1 & a_2^1 & 0 \\ 0 & 0 & a_3^1 \\ 0 & 0 & a_4^1 \end{pmatrix}$$

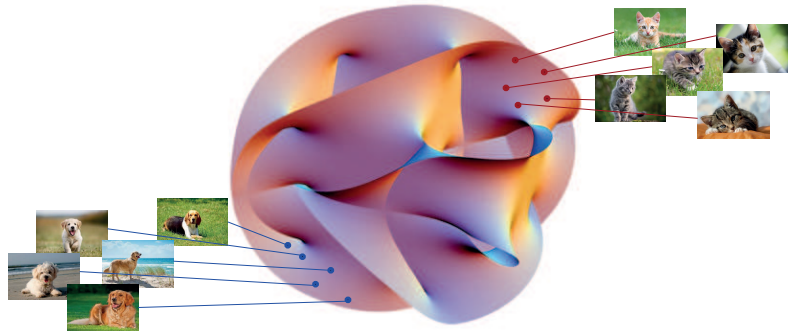
$$A_2 = \begin{pmatrix} a_1^2 & a_2^2 & 0 \\ 0 & 0 & a_3^2 \end{pmatrix}$$



Training of Deep Neural Networks

High-Level Set Up:

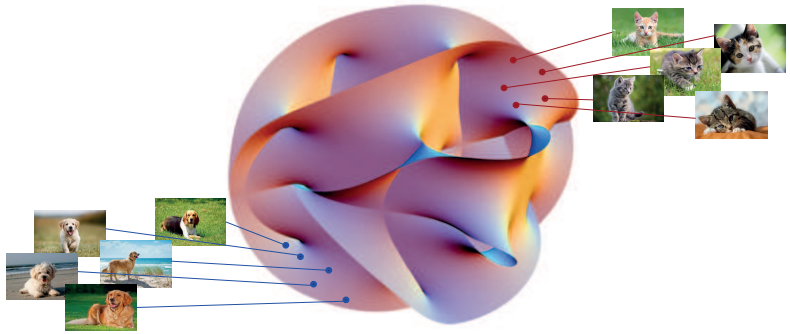
- Samples $(x_i, f(x_i))_{i=1}^m$ of a function such as $f : \mathcal{M} \to \{1, 2, \dots, K\}$.



Training of Deep Neural Networks

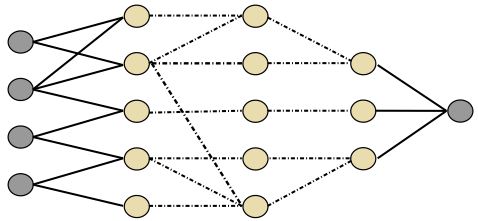
High-Level Set Up:

- Samples $(x_i, f(x_i))_{i=1}^m$ of a function such as $f : \mathcal{M} \to \{1, 2, \dots, K\}$.



- Select an architecture of a deep neural network, i.e., a choice of d, L, $(N_\ell)_{\ell=1}^L$, and σ.

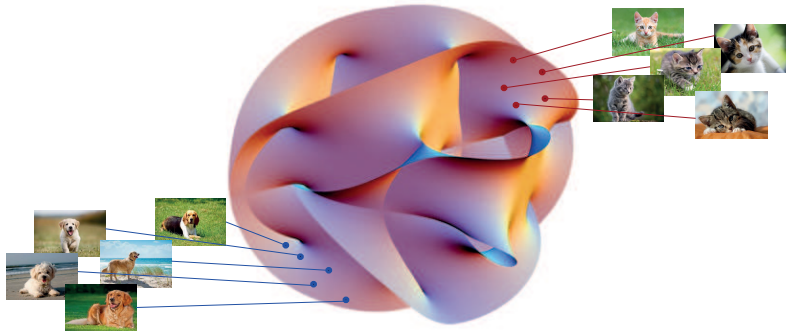
  Sometimes selected entries of the matrices $(A_\ell)_{\ell=1}^L$, i.e., weights, are set to zero at this point.



Training of Deep Neural Networks

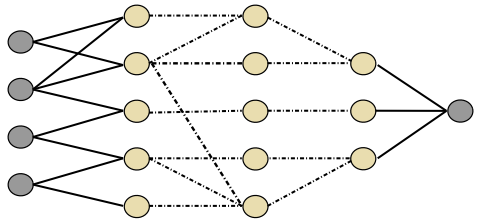
High-Level Set Up:

- Samples $(x_i, f(x_i))_{i=1}^m$ of a function such as $f : \mathcal{M} \to \{1, 2, \dots, K\}$.



- Select an architecture of a deep neural network, i.e., a choice of d, L, $(N_\ell)_{\ell=1}^L$, and σ.

  Sometimes selected entries of the matrices $(A_\ell)_{\ell=1}^L$, i.e., weights, are set to zero at this point.



- Learn the affine-linear functions $(W_\ell)_{\ell=1}^L = (A_\ell \cdot + b_\ell)_{\ell=1}^L$ by

$$\min_{A_\ell, b_\ell} \sum_{i=1}^{m} \mathcal{L}(\Phi_{A_\ell, b_\ell}(x_i), f(x_i)) + \lambda \mathcal{R}(A_\ell, b_\ell)$$

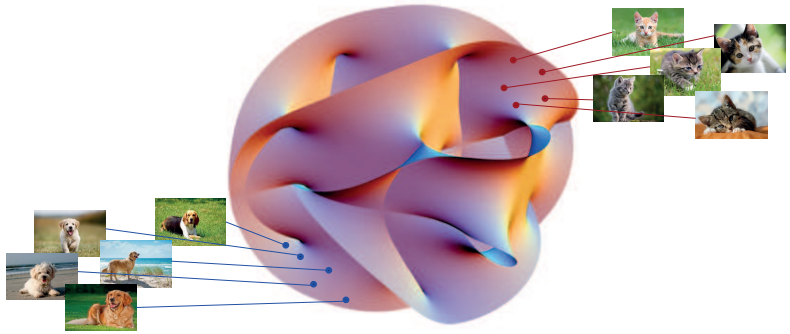
  yielding the network $\Phi_{A_\ell, b_\ell} : \mathbb{R}^d \to \mathbb{R}^{N_L}$,

$$\Phi_{A_\ell, b_\ell}(x) = W_L \sigma(W_{L-1} \sigma(\dots \sigma(W_1(x))).$$

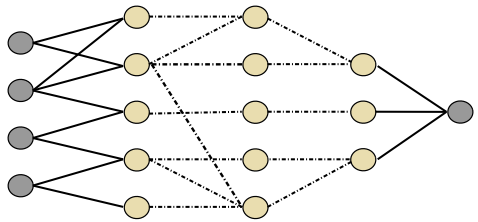
This is often done by stochastic gradient descent.

Training of Deep Neural Networks

High-Level Set Up:



- Samples $(x_i, f(x_i))_{i=1}^m$ of a function such as $f : \mathcal{M} \to \{1, 2, \dots, K\}$.



- Select an architecture of a deep neural network, i.e., a choice of d, L, $(N_\ell)_{\ell=1}^L$, and σ.

  Sometimes selected entries of the matrices $(A_\ell)_{\ell=1}^L$, i.e., weights, are set to zero at this point.

- Learn the affine-linear functions $(W_\ell)_{\ell=1}^L = (A_\ell \cdot + b_\ell)_{\ell=1}^L$ by

$$\min_{A_\ell, b_\ell} \sum_{i=1}^{m} \mathcal{L}(\Phi_{A_\ell, b_\ell}(x_i), f(x_i)) + \lambda \mathcal{R}(A_\ell, b_\ell)$$

  yielding the network $\Phi_{A_\ell, b_\ell} : \mathbb{R}^d \to \mathbb{R}^{N_L}$,

$$\Phi_{A_\ell, b_\ell}(x) = W_L \sigma(W_{L-1} \sigma(\dots \sigma(W_1(x))).$$

This is often done by stochastic gradient descent.

Goal: $\Phi \approx f$

Second Appearance of Neural Networks

Key Observations by Y. LeCun et al. (around 2000):

- Drastic improvement of computing power.
  - $\rightsquigarrow$ *Networks with hundreds of layers can be trained.*
  - $\rightsquigarrow$ *Deep Neural Networks!*
- Age of Data starts.
  - $\rightsquigarrow$ *Vast amounts of training data is available.*

Second Appearance of Neural Networks

Key Observations by Y. LeCun et al. (around 2000):

- Drastic improvement of computing power.
  $\rightsquigarrow$ *Networks with hundreds of layers can be trained.*
  $\rightsquigarrow$ *Deep Neural Networks!*
- Age of Data starts.
  $\rightsquigarrow$ *Vast amounts of training data is available.*

Current Situation:

- Setting up a deep neural network for a particular application is more or less trail-and-error and based on experience.
- Training a deep neural network is very unpredictable.
- Almost no knowledge about why a deep neural network mades a decision.

Danger of Deep Neural Networks?

AI researchers allege that machine learning is alchemy:

"Ali Rahimi, a researcher in artificial intelligence (AI) at Google in San Francisco, California, took a swipe at his field last December and received a 40-second ovation for it. Speaking at an AI conference, Rahimi charged that machine learning algorithms, in which computers learn through trial and error, have become a form of "alchemy". Researchers, he said, do not know why some algorithms work and others don't, nor do they have rigorous criteria for choosing one AI architecture over another...."

"For example, he says, they adopt pet methods to tune their AIs' "learning rates" how much an algorithm corrects itself after each mistake –without understanding why one is better than others. In other cases, AI researchers training their algorithms are simply stumbling in the dark. For example, they implement what's called "stochastic gradient descent" in order to optimize an algorithm's parameters for the lowest possible failure rate. Yet despite thousands of academic papers on the subject, and countless ways of applying the method, the process still relies on trial and error...."

Science (May 2018)

Fundamental Questions concerning Deep Neural Networks

- *Expressivity:*
  - How powerful is the network architecture?
  - Can it indeed represent the correct functions?
- *Learning:*
  - Why does the current learning algorithm produce anything reasonable?
  - What are good starting values?
- *Generalization:*
  - Why do deep neural networks perform that well on data sets, which do not belong to the input-output pairs from a training set?
  - What impact has the depth of the network?
- *Explainability:*
  - Why did a trained deep neural network reach a certain decision?
  - Which components of the input do contribute most?

Fundamental Questions concerning Deep Neural Networks

- *Expressivity:*
  - How powerful is the network architecture?
  - Can it indeed represent the correct functions?

  $\rightsquigarrow$ *Applied Harmonic Analysis, Approximation Theory, ...*
- *Learning:*
  - Why does the current learning algorithm produce anything reasonable?
  - What are good starting values?

- *Generalization:*
  - Why do deep neural networks perform that well on data sets, which do not belong to the input-output pairs from a training set?
  - What impact has the depth of the network?

- *Explainability:*
  - Why did a trained deep neural network reach a certain decision?
  - Which components of the input do contribute most?

Fundamental Questions concerning Deep Neural Networks

- *Expressivity:*
  - How powerful is the network architecture?
  - Can it indeed represent the correct functions?

  $\rightsquigarrow$ *Applied Harmonic Analysis, Approximation Theory, ...*
- *Learning:*
  - Why does the current learning algorithm produce anything reasonable?
  - What are good starting values?

  $\rightsquigarrow$ *Differential Geometry, Optimal Control, Optimization, ...*
- *Generalization:*
  - Why do deep neural networks perform that well on data sets, which do not belong to the input-output pairs from a training set?
  - What impact has the depth of the network?

- *Explainability:*
  - Why did a trained deep neural network reach a certain decision?
  - Which components of the input do contribute most?

Fundamental Questions concerning Deep Neural Networks

- *Expressivity:*
  - How powerful is the network architecture?
  - Can it indeed represent the correct functions?

  $\rightsquigarrow$ *Applied Harmonic Analysis, Approximation Theory, ...*
- *Learning:*
  - Why does the current learning algorithm produce anything reasonable?
  - What are good starting values?

  $\rightsquigarrow$ *Differential Geometry, Optimal Control, Optimization, ...*
- *Generalization:*
  - Why do deep neural networks perform that well on data sets, which do not belong to the input-output pairs from a training set?
  - What impact has the depth of the network?

  $\rightsquigarrow$ *Learning Theory, Optimization, Statistics, ...*
- *Explainability:*
  - Why did a trained deep neural network reach a certain decision?
  - Which components of the input do contribute most?

Fundamental Questions concerning Deep Neural Networks

- *Expressivity:*
  - How powerful is the network architecture?
  - Can it indeed represent the correct functions?

  $\rightsquigarrow$ *Applied Harmonic Analysis, Approximation Theory, ...*
- *Learning:*
  - Why does the current learning algorithm produce anything reasonable?
  - What are good starting values?

  $\rightsquigarrow$ *Differential Geometry, Optimal Control, Optimization, ...*
- *Generalization:*
  - Why do deep neural networks perform that well on data sets, which do not belong to the input-output pairs from a training set?
  - What impact has the depth of the network?

  $\rightsquigarrow$ *Learning Theory, Optimization, Statistics, ...*
- *Explainability:*
  - Why did a trained deep neural network reach a certain decision?
  - Which components of the input do contribute most?

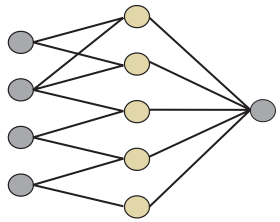
  $\rightsquigarrow$ *Information Theory, Uncertainty Quantification, ...*

A Glimpse into the Theory

Expressivity: Universality

Universal Approximation Theorem (Cybenko/Hornik/Pinkus, 1989–1999):
Let $\sigma : \mathbb{R} \to \mathbb{R}$ be continuous, but not a polynomial. Also, fix $d \geq 1$, $L = 2$, $N_L \geq 1$, and a compact set $K \subseteq \mathbb{R}^d$. Then, for any continuous $f : \mathbb{R}^d \to \mathbb{R}^{N_L}$ and every $\varepsilon > 0$, there exist $N \in \mathbb{N}$, $a_k, b_k \in \mathbb{R}, w_k \in \mathbb{R}^d$ such that

$$\sup_{x \in K} \Big| \sum_{k=1}^{N} a_k \sigma(\langle w_k, x \rangle) - f(x) \Big| \leq \varepsilon.$$

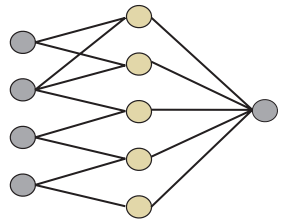


Every continuous function can be approximated up to $\varepsilon > 0$ by a neural network with one hidden layer and $O(N)$ neurons.

Expressivity: Universality

Universal Approximation Theorem (Cybenko/Hornik/Pinkus, 1989–1999):
Let $\sigma : \mathbb{R} \to \mathbb{R}$ be continuous, but not a polynomial. Also, fix $d \geq 1$, $L = 2$, $N_L \geq 1$, and a compact set $K \subseteq \mathbb{R}^d$. Then, for any continuous $f : \mathbb{R}^d \to \mathbb{R}^{N_L}$ and every $\varepsilon > 0$, there exist $N \in \mathbb{N}$, $a_k, b_k \in \mathbb{R}$, $w_k \in \mathbb{R}^d$ such that

$$\sup_{x \in K} \Big| \sum_{k=1}^{N} a_k \sigma(\langle w_k, x \rangle) - f(x) \Big| \leq \varepsilon.$$



Every continuous function can be approximated up to $\varepsilon > 0$ by a neural network with one hidden layer and $O(N)$ neurons.

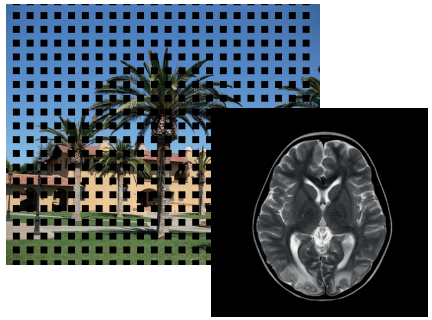
Additional approximation results aiming to understand the impact of...

- the activation function,
- the chosen function class,
- the depth of the neural network,
- ...

Solving Inverse Problems

Examples:

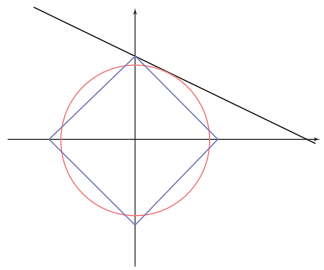
- Denoising.
- Feature Extraction.
- Inpainting.
- Magnetic Resonance Tomography.
- ...



Sparse Regularization (Compressed Sensing):

Given an ill-posed inverse problem $Kx = y$, where $K : X \to Y$ and x is known to be sparsely representable by an ONB/frame $(\sigma_\eta)_\eta$, an approximate solution $x^\alpha \in X$, $\alpha > 0$, can be determined by

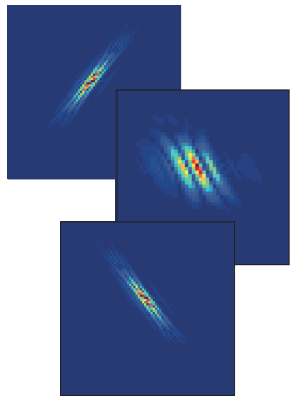
$$\min_{\tilde{x}} \|K\tilde{x} - y\|^2 + \alpha \|(\langle \tilde{x}, \sigma_\eta \rangle)_\eta\|_1.$$



Shearlets

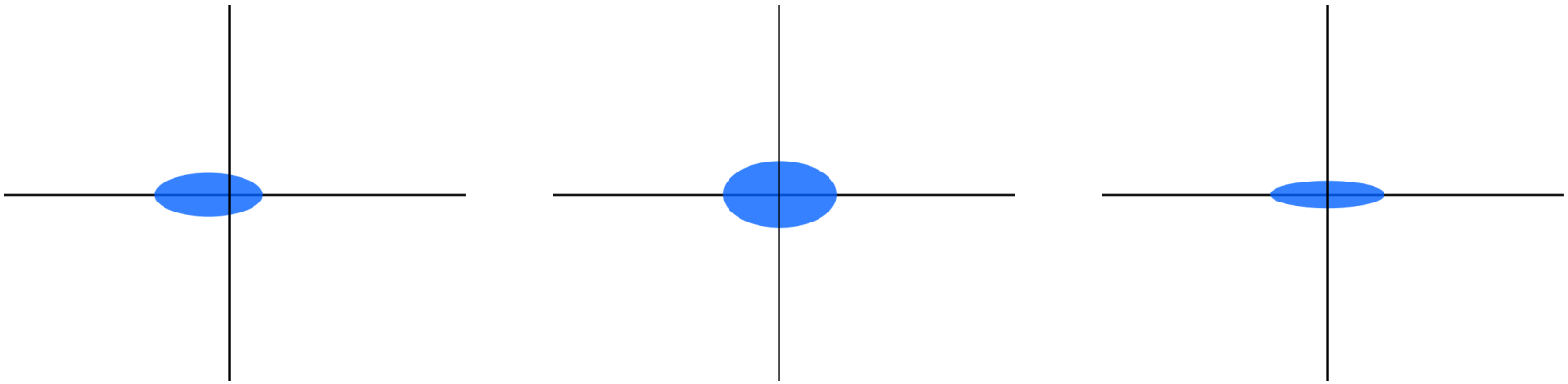
Shearlets (Kutyniok, Labate; 2006):

$$A_j := \begin{pmatrix} 2^j & 0 \\ 0 & 2^{j/2} \end{pmatrix}, \qquad S_k := \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix}, \quad j, k \in \mathbb{Z}.$$



Then

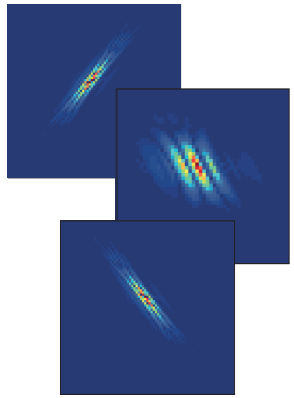
$$\psi_{j,k,m} := 2^{\frac{3j}{4}} \psi(S_k A_j \cdot - m).$$



Shearlets

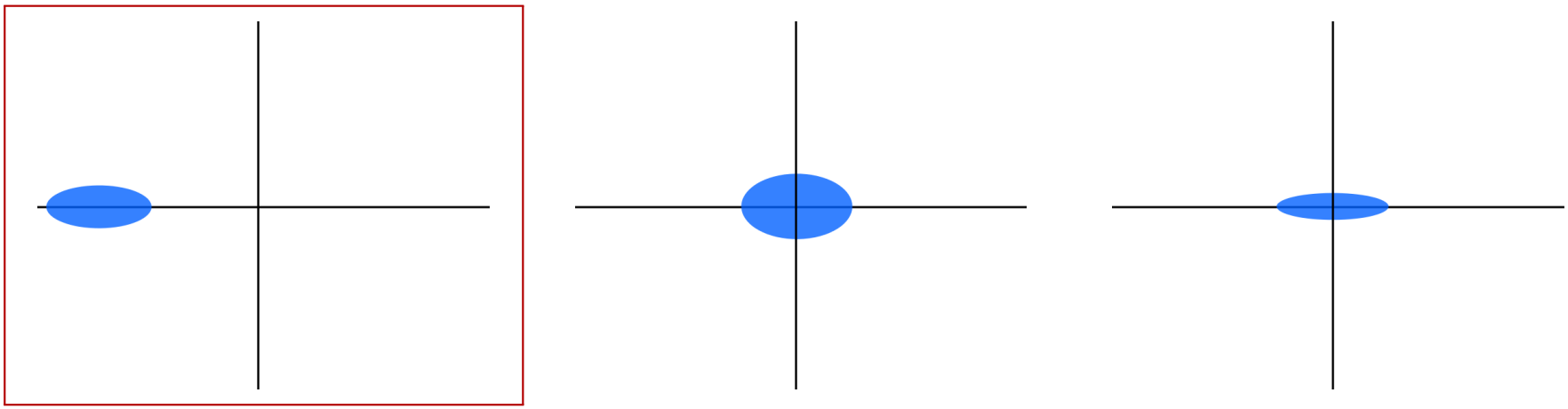
Shearlets (Kutyniok, Labate; 2006):

$$A_j := \begin{pmatrix} 2^j & 0 \\ 0 & 2^{j/2} \end{pmatrix}, \qquad S_k := \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix}, \quad j, k \in \mathbb{Z}.$$



Then

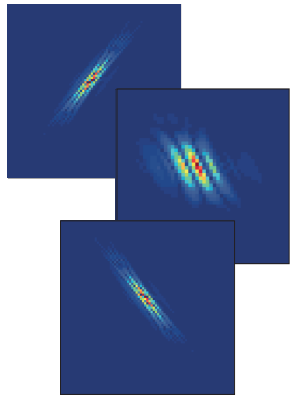
$$\psi_{j,k,m} := 2^{\frac{3j}{4}} \psi(S_k A_j \cdot - m).$$



Shearlets

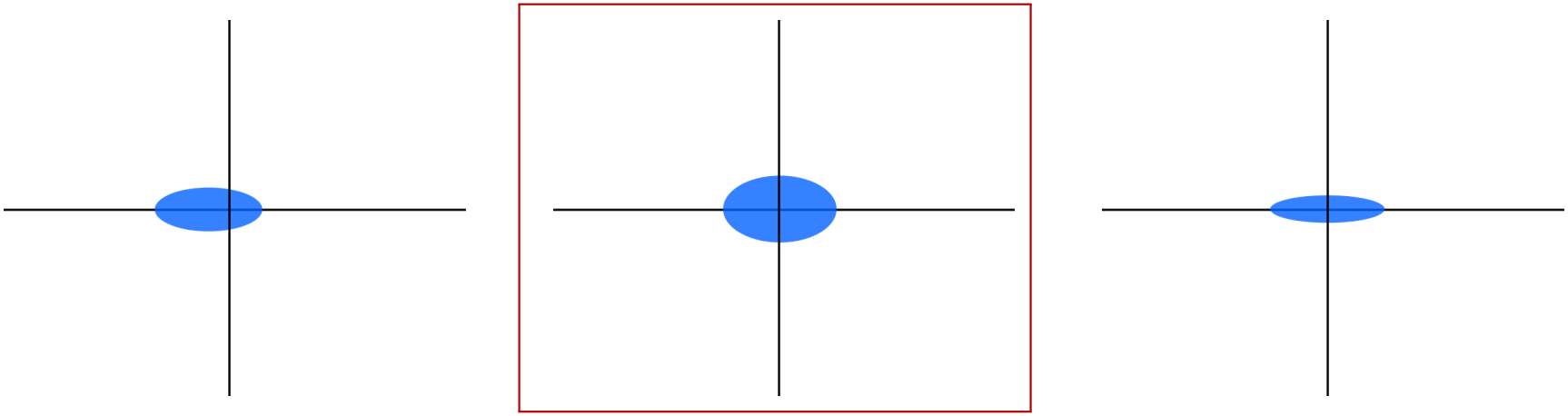
Shearlets (Kutyniok, Labate; 2006):

$$A_j := \begin{pmatrix} 2^j & 0 \\ 0 & 2^{j/2} \end{pmatrix}, \qquad S_k := \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix}, \quad j, k \in \mathbb{Z}.$$



Then

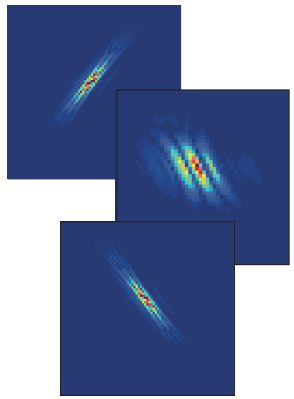
$$\psi_{j,k,m} := 2^{\frac{3j}{4}} \psi(S_k A_j \cdot - m).$$



Shearlets

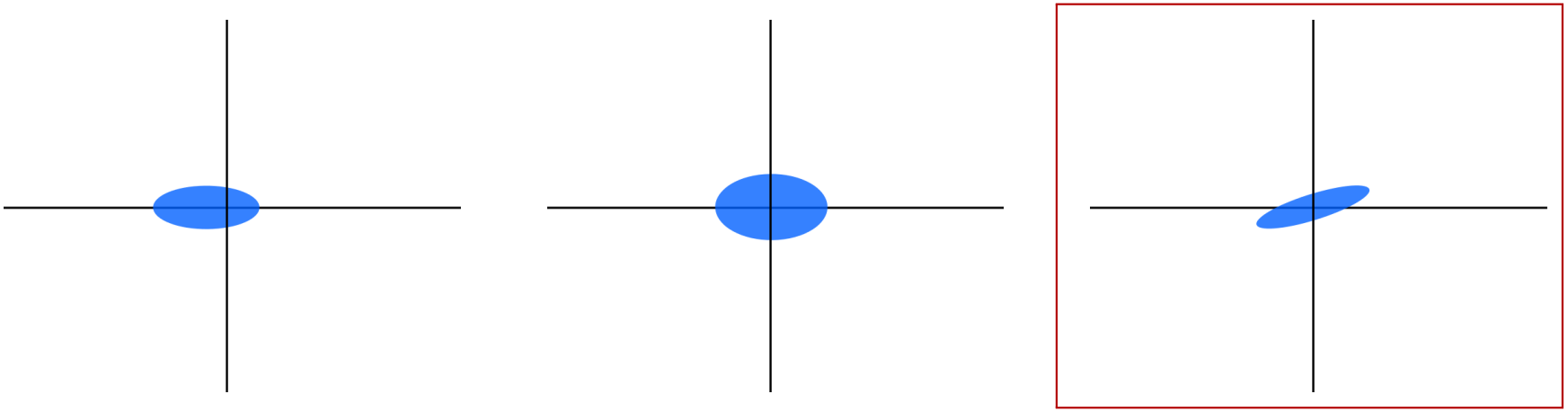
Shearlets (Kutyniok, Labate; 2006):

$$A_j := \begin{pmatrix} 2^j & 0 \\ 0 & 2^{j/2} \end{pmatrix}, \qquad S_k := \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix}, \quad j, k \in \mathbb{Z}.$$



Then

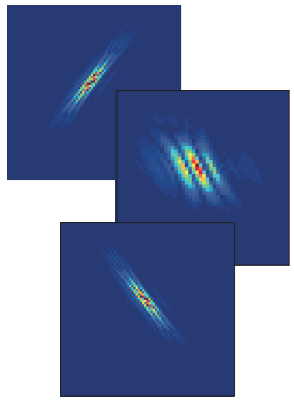
$$\psi_{j,k,m} := 2^{\frac{3j}{4}} \psi(S_k A_j \cdot - m).$$



Shearlets

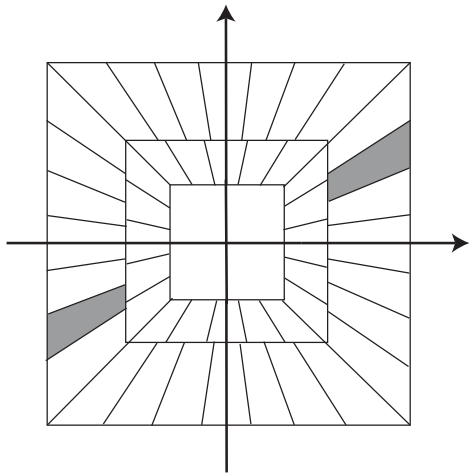
Shearlets (Kutyniok, Labate; 2006):

$$A_j := \begin{pmatrix} 2^j & 0 \\ 0 & 2^{j/2} \end{pmatrix}, \qquad S_k := \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix}, \quad j, k \in \mathbb{Z}.$$



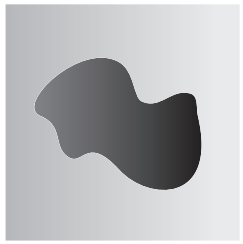
Then

$$\psi_{j,k,m} := 2^{\frac{3j}{4}} \psi(S_k A_j \cdot - m).$$



Shearlets are Optimal

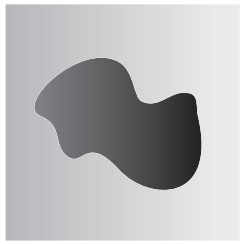
Model of Images (Donoho; 2001):
"Cartoon-functions are functions governed by a discontinuity curve."



Theorem (Kutyniok, Lim; 2011):
"Shearlets fulfill the optimal approximation rate for cartoon-functions."

Shearlets are Optimal

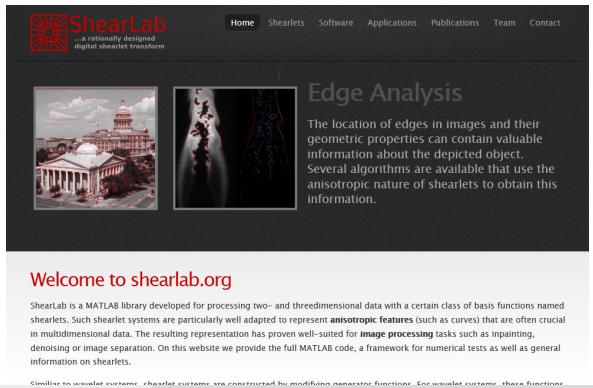
Model of Images (Donoho; 2001):
"Cartoon-functions are functions governed by a discontinuity curve."



Theorem (Kutyniok, Lim; 2011):
"Shearlets fulfill the optimal approximation rate for cartoon-functions."

2D&3D (parallelized) Fast Shearlet Transform (`www.ShearLab.org`):

- Matlab *(Kutyniok, Lim, Reisenhofer; 2013)*
- Julia *(Loarca; 2017)*
- Python *(Look; 2018)*
- Tensorflow *(Loarca; 2019)*



Optimally Sparsely Connected Neural Networks

Theorem (Bölcskei, Grohs, Kutyniok, and Petersen; 2017): Let ρ be an admissible smooth activation function, and let $\varepsilon > 0$. Then there exist $C_\varepsilon > 0$ such that, for all cartoon-like functions f and $N \in \mathbb{N}$, we can construct a neural network Φ with $O(N)$ connections *(This is minimal!)* and 3 layers satisfying

$$\|f - \Phi\|_{L^2(\mathbb{R}^2)} \leq C_\varepsilon N^{-1-\varepsilon}.$$

Remark: The topology and quantized weights of this network can be stored with $C \cdot N \cdot \text{polylog}(N)$ bits.

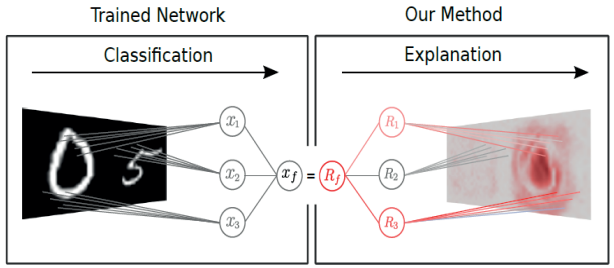
Function classes which are optimal representable by affine systems are also optimally effectively approximated by memory-efficient neural networks with a parallel architecture!

Explainability

Main Questions:

Given a trained deep neural network.

- Which input features contribute most to the decision?
- How can the outcome be explained?

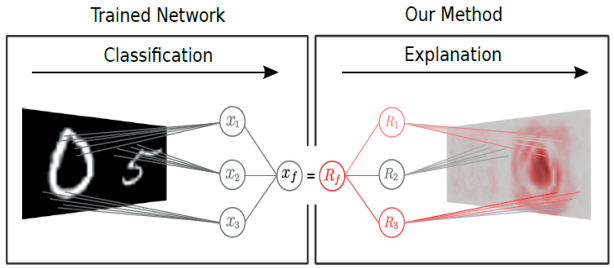


Explainability

Main Questions:

Given a trained deep neural network.

- Which input features contribute most to the decision?
- How can the outcome be explained?



Theorem (Wäldchen, Macdonald, Hauch, Kutyniok; 2019):

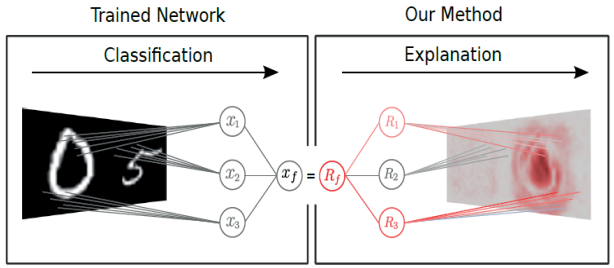
"The (optimization) task of determining the most relevant input signal components for a Boolean classifier decision for binary signals is NN^{PP}-complete and NP-hard to approximate."

Explainability

Main Questions:

Given a trained deep neural network.

- Which input features contribute most to the decision?
- How can the outcome be explained?

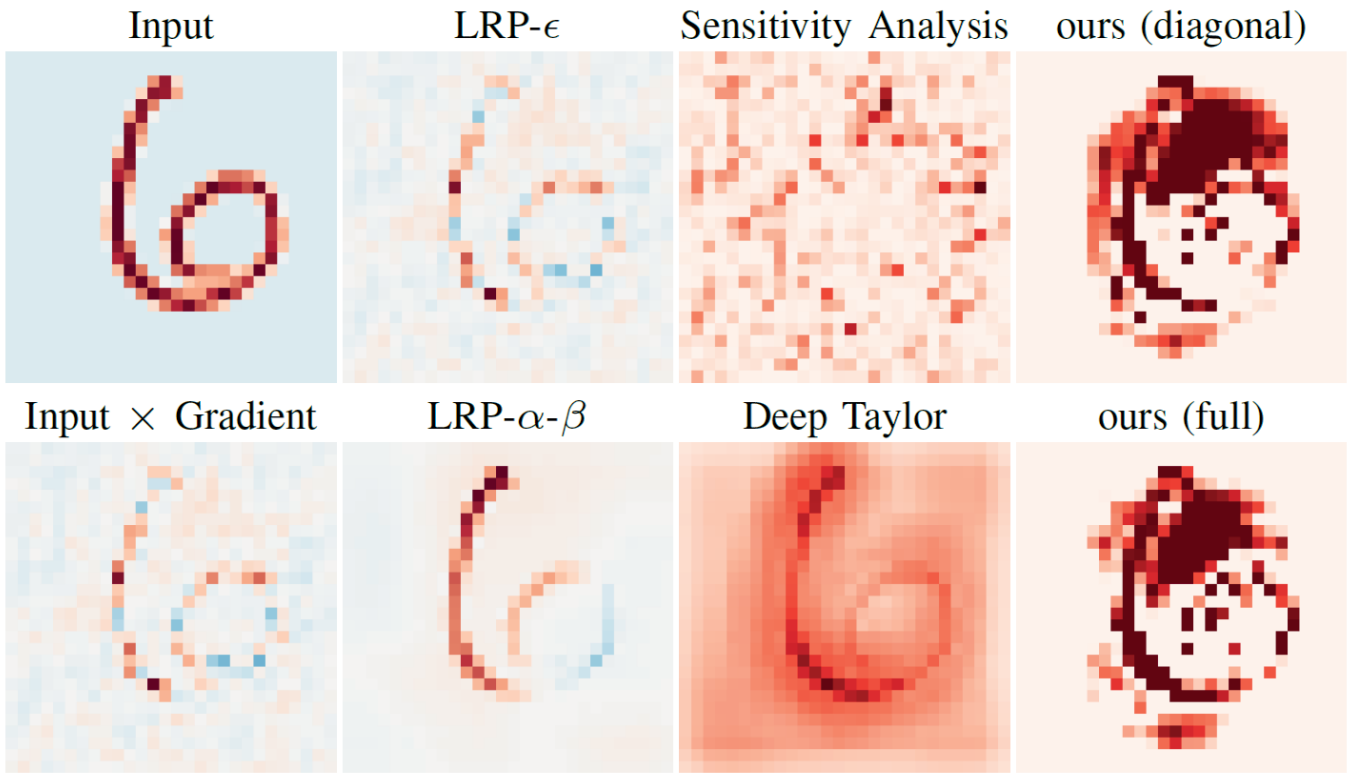


Theorem (Wäldchen, Macdonald, Hauch, Kutyniok; 2019):

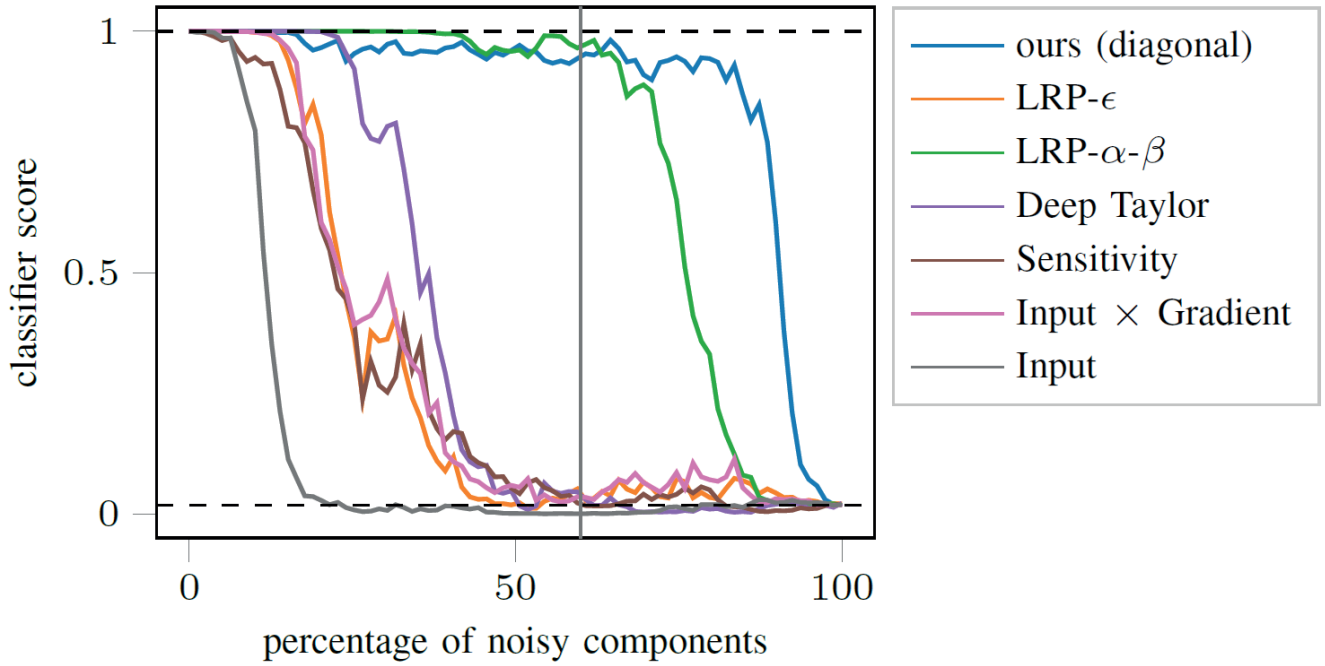
"The (optimization) task of determining the most relevant input signal components for a Boolean classifier decision for binary signals is NN^{PP}-complete and NP-hard to approximate."

$\rightsquigarrow$ *Solution strategy of a relaxed optimization problem based on assumed density filtering (Wäldchen, Macdonald, Hauch, Kutyniok; 2019)!*

Numerical Results, I (Wäldchen, Macdonald, Hauch, Kutyniok; 2019)



Numerical Results, II (Wäldchen, Macdonald, Hauch, Kutyniok; 2019)



Taking the Best out of Both Worlds

From Data-Driven to Model-Based Approaches

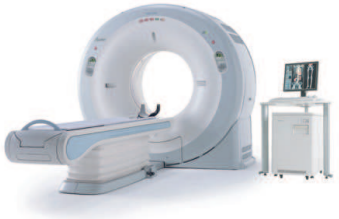
Problems, Viewpoints and Solution Strategies:

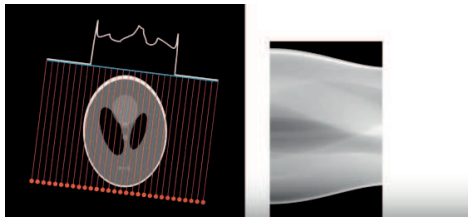
- Pure data-driven approaches.
  Detect structural components in data sets!
- Machine learning with physical constraints.
  Insert physical information in machine learning algorithm!
- Parametric differential equations.
  Learn parameters from given data sets!
- Data assimilation.
  Combine sparse data with physical model to generate a general model!
- Data analysis on simulation data.
  Study simulation generated data in search of underlying laws!

Model based & data driven approaches:
Only learn what needs to be learned!

Computed Tomography (CT)

Computed Tomography (CT):

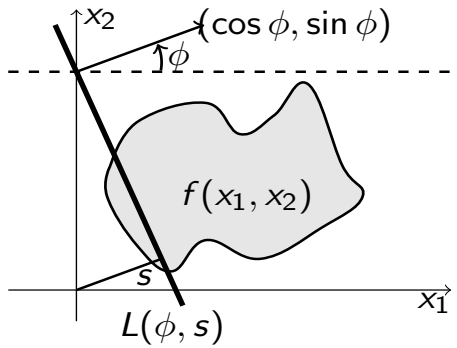




A CT scanner samples the *Radon transform*

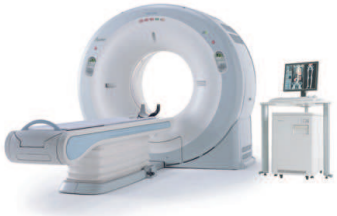
$$\mathcal{R}f(\phi, s) = \int_{L(\phi,s)} f(x) dS(x),$$

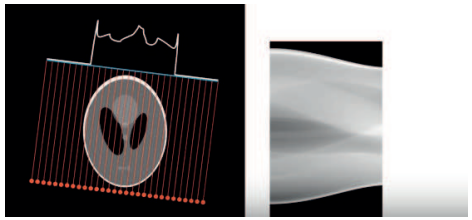
for $L(\phi, s) = \left\{x \in \mathbb{R}^2 : x_1 \cos(\phi) + x_2 \sin(\phi) = s\right\}$,
$\phi \in [-\pi/2, \pi/2)$, and $s \in \mathbb{R}$.



Computed Tomography (CT)

Computed Tomography (CT):

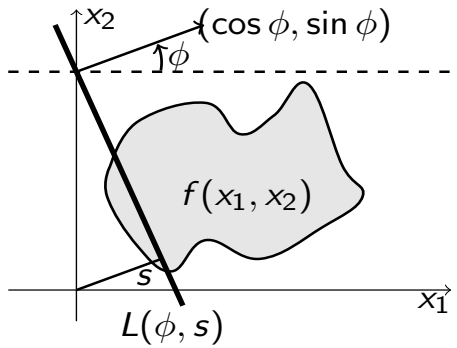




A CT scanner samples the *Radon transform*

$$\mathcal{R}f(\phi, s) = \int_{L(\phi,s)} f(x) dS(x),$$

for $L(\phi, s) = \left\{x \in \mathbb{R}^2 : x_1 \cos(\phi) + x_2 \sin(\phi) = s\right\}$,
$\phi \in [-\pi/2, \pi/2)$, and $s \in \mathbb{R}$.



Limited-Angle Computed Tomography:

Challenging inverse problem if $\mathcal{R}f(\cdot, s)$ is only sampled on $[-\phi, \phi] \subset [-\pi/2, \pi/2)$.

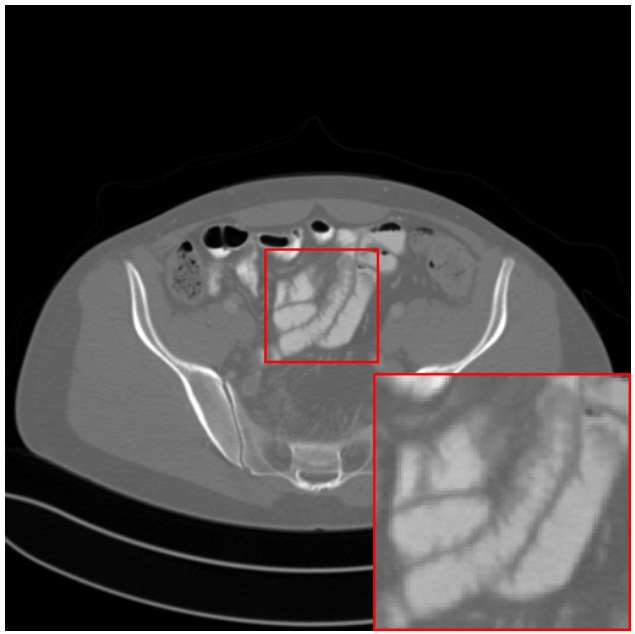
Applications: Dental CT, breast tomosynthesis, electron tomography,...

Model-Based Approaches Fail

Sparse Regularization with Shearlets $(\psi_{j,k,m})_{j,k,m}$:

$$\text{argmin}_f \left[\|\mathcal{R}f - g\|^2 + \alpha \cdot \|(\langle f, \psi_{j,k,m} \rangle)_{j,k,m}\|_1 \right].$$

Clinical Data (60° missing wedge):



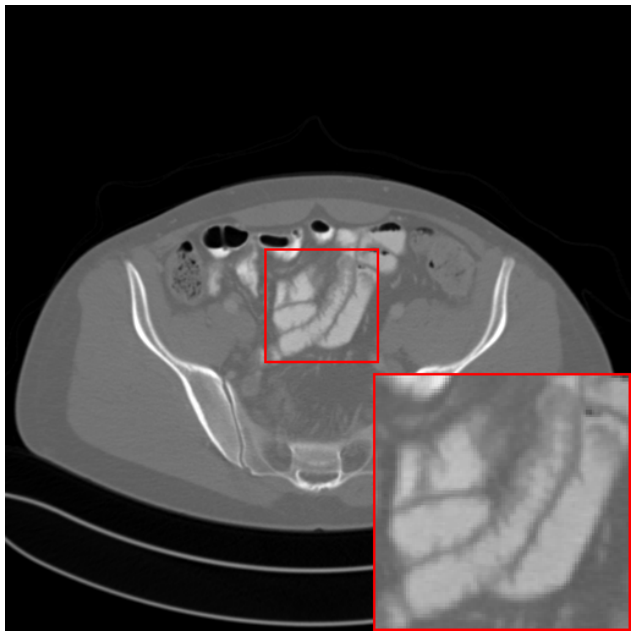
Original Image

Model-Based Approaches Fail

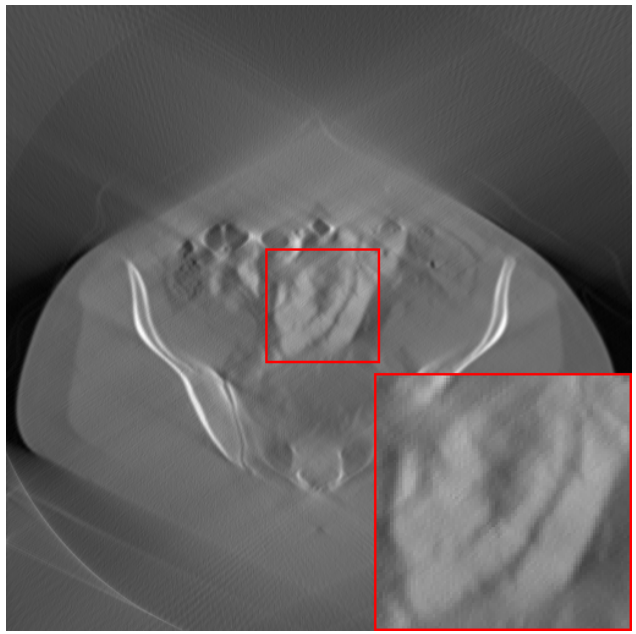
Sparse Regularization with Shearlets $(\psi_{j,k,m})_{j,k,m}$:

$$\text{argmin}_f \left[\|\mathcal{R}f - g\|^2 + \alpha \cdot \|(\langle f, \psi_{j,k,m} \rangle)_{j,k,m}\|_1 \right].$$

Clinical Data (60° missing wedge):



Original Image



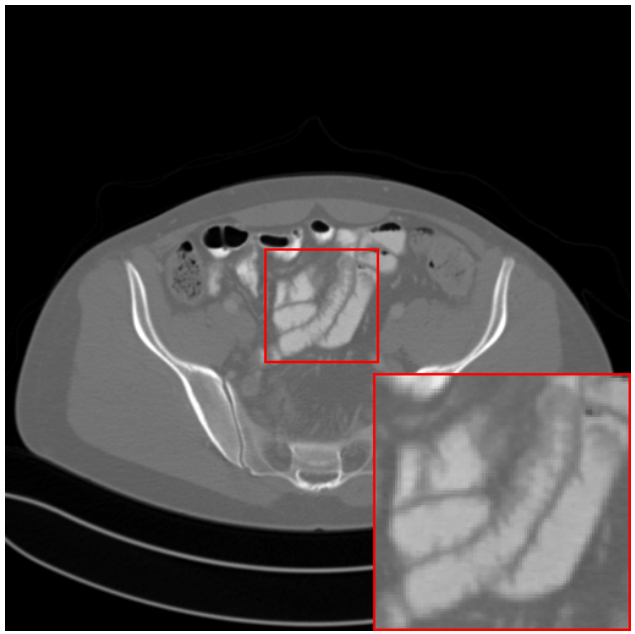
Filtered Backprojection

Model-Based Approaches Fail

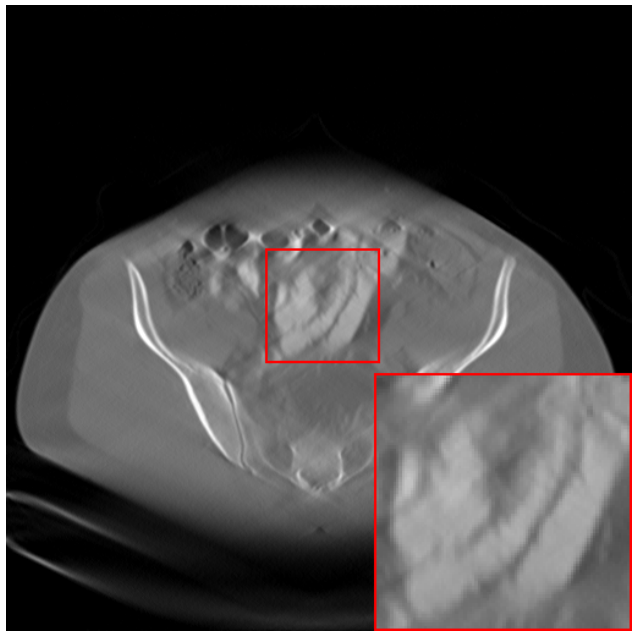
Sparse Regularization with Shearlets $(\psi_{j,k,m})_{j,k,m}$:

$$\text{argmin}_f \left[\|\mathcal{R}f - g\|^2 + \alpha \cdot \|(\langle f, \psi_{j,k,m} \rangle)_{j,k,m}\|_1 \right].$$

Clinical Data (60° missing wedge):



Original Image

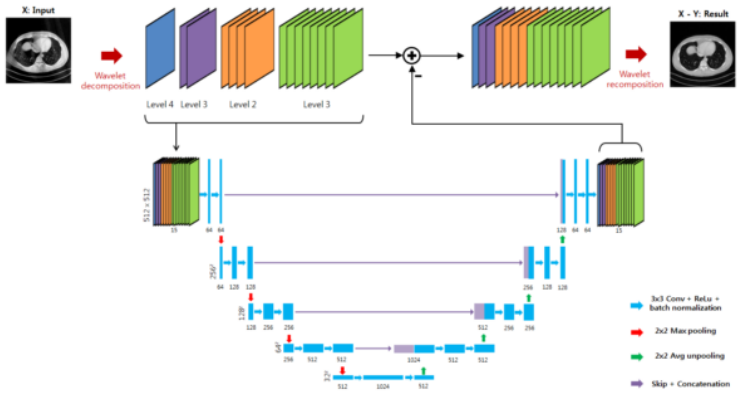


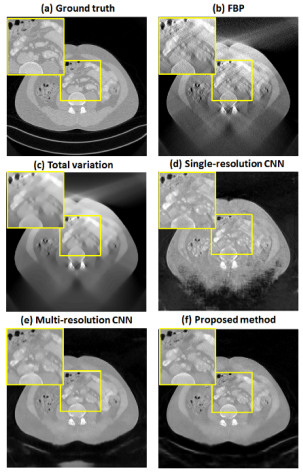
Sparse Regularization with Shearlets

Deep Learning for Inverse Problems

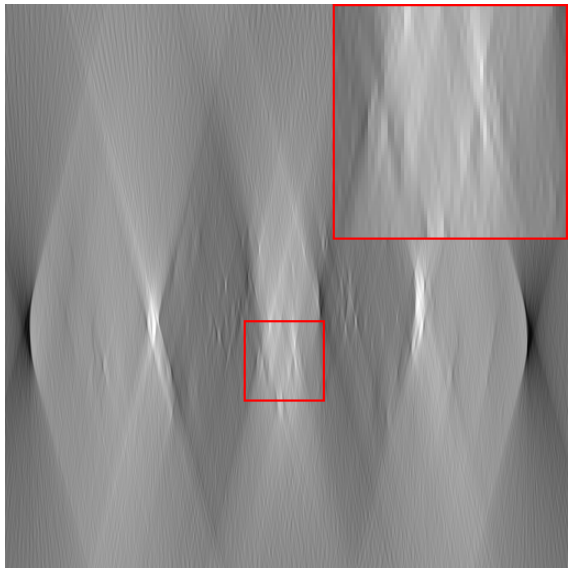
Incomplete List:

- On FBP:
  - [Kang et al., 2017]: contourlets of FBP + U-net, 2nd place Mayo low-dose challenge & many more works from this group!
  - [Zhang et al., 2016]: 2-layer network on FBP
  - [Jin et al., 2017]: U-Net on FBP
- Incorporating forward model via optimization scheme:
  - [Hammernik et al., 2017]: learning weights for FBP, then filtering with gradient steps
  - [Meinhardt et al., 2017]: learning proximal operators
  - [Adler et al., 2017]: learned primal dual
- Deep Learning Approach for Limited-Angle CT [Gu & Ye, 2017]



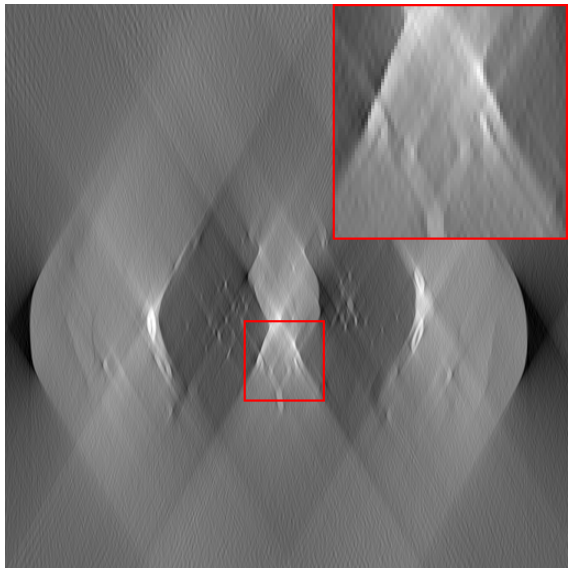


Zooming in on the Recovery Problem



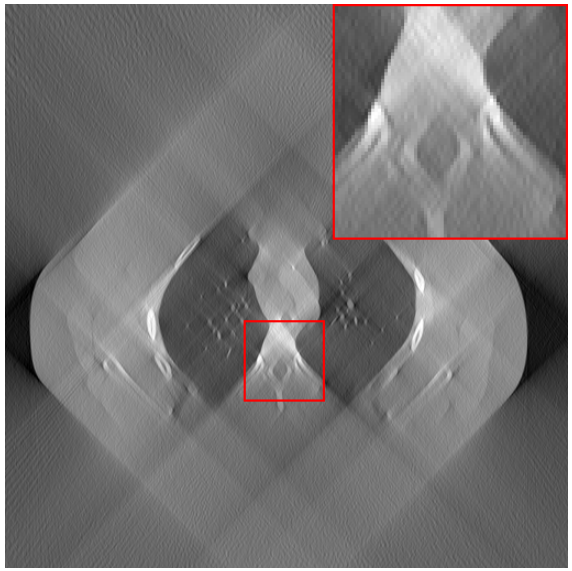
$\phi = 15°$, filtered backprojection (FBP)

Zooming in on the Recovery Problem



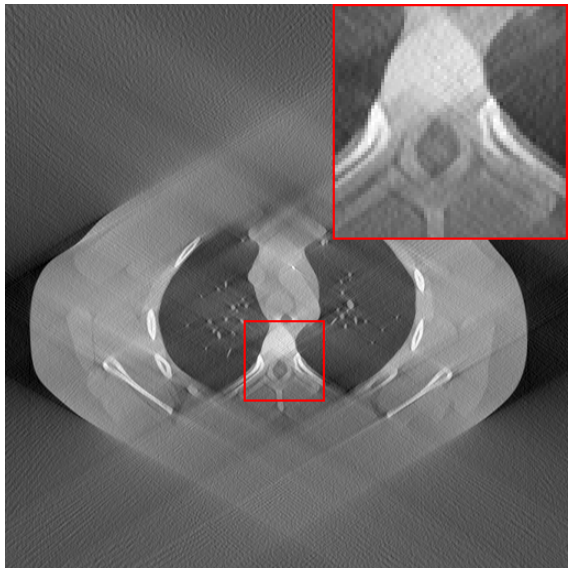
$\phi = 30°$, filtered backprojection (FBP)

Zooming in on the Recovery Problem



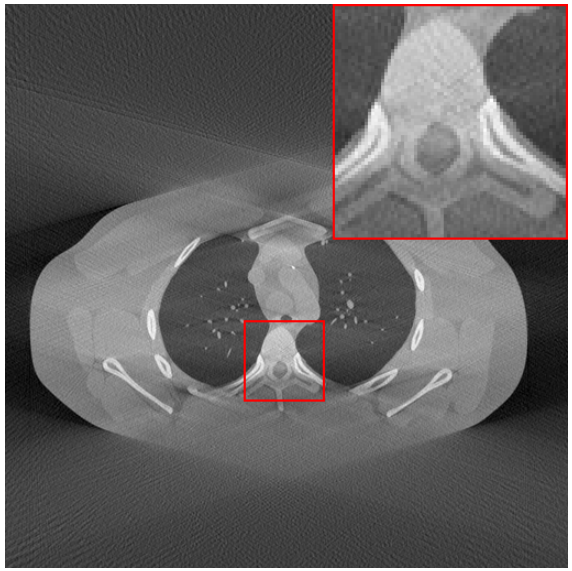
$\phi = 45°$, filtered backprojection (FBP)

Zooming in on the Recovery Problem



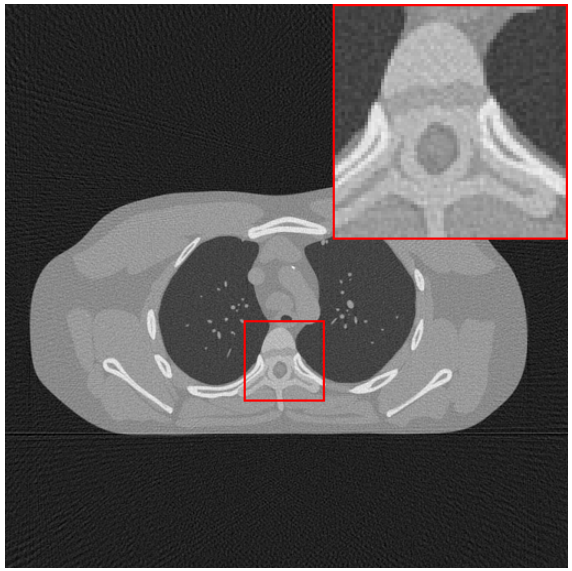
$\phi = 60°$, filtered backprojection (FBP)

Zooming in on the Recovery Problem



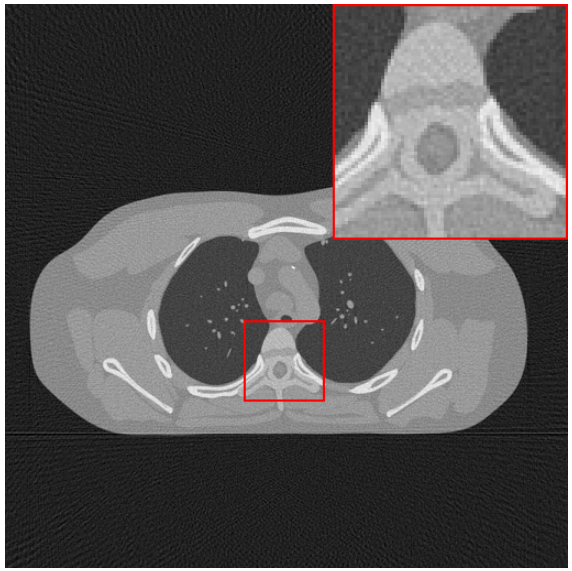
$\phi = 75°$, filtered backprojection (FBP)

Zooming in on the Recovery Problem



$\phi = 90°$, filtered backprojection (FBP)

Zooming in on the Recovery Problem



$\phi = 90°$, filtered backprojection (FBP)

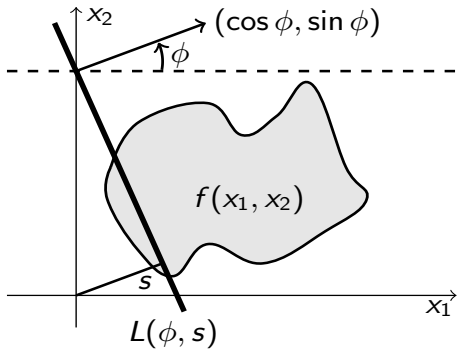
Some Observations:

- Only certain boundaries/features seem to be *"visible"*!
- Missing wedge creates artifacts!
- Highly ill-posed inverse problem!

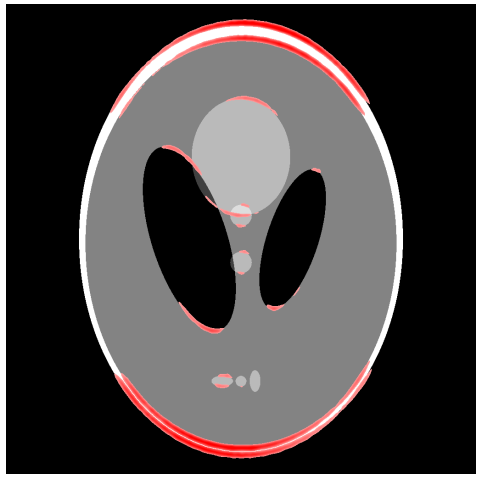
Visibility in CT

Theorem ([Quinto, 1993]): Let $L_0 = L(\phi_0, s_0)$ be a line in the plane. Let $(x_0, \xi_0) \in \mathrm{WF}(f)$ such that $x_0 \in L_0$ and ξ_0 is a normal vector to L_0.

- The singularity of f at (x_0, ξ_0) causes a unique singularity in $\mathcal{R}f$ at (ϕ_0, s_0).
- Singularities of f not tangent to $L(\phi_0, s_0)$ do not cause singularities in $\mathcal{R}f$ at (ϕ_0, s_0).



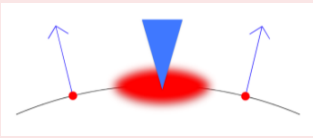
"visible": singularities tangent to sampled lines



"invisible": singularities not tangent to sampled lines

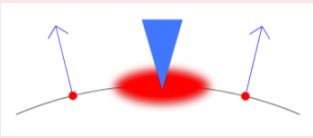
Shearlets can Help

Key Idea: Filling the missing angle is an inpainting problem of the wavefront set!

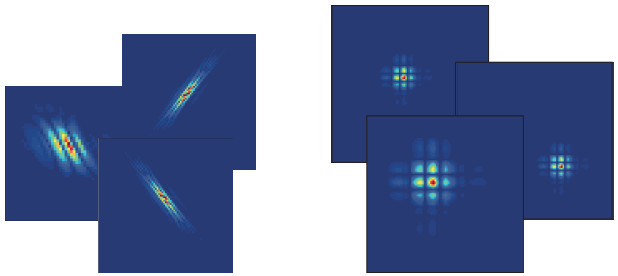


Shearlets can Help

Key Idea: Filling the missing angle is an inpainting problem of the wavefront set!



Theorem (Kutyniok, Labate, 2006): "Shearlets can identify the wavefront set at fine scales."



More Precisely:

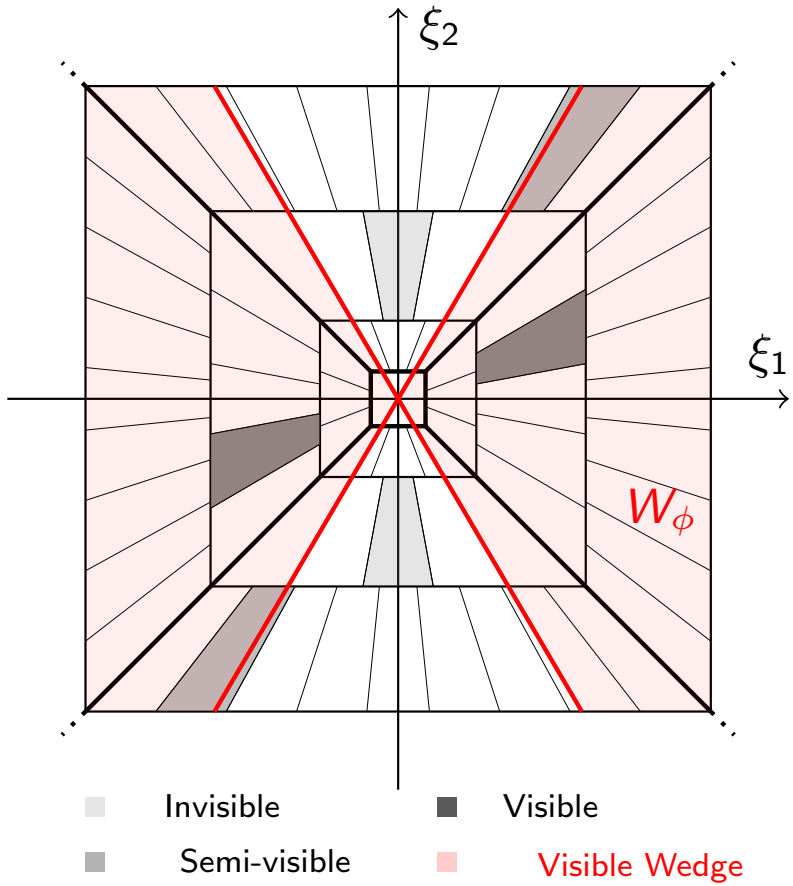
- Continuous Shearlet Transform:

$$L^2(\mathbb{R}^2) \ni f \mapsto \mathcal{SH}_\psi f(a,s,t) = \langle f, \psi_{a,s,t} \rangle, \quad (a,s,t) \in \mathbb{R}_+ \times \mathbb{R} \times \mathbb{R}^2.$$

- Resolution of Wavefront Sets (simplified from [Kutyniok & Labate, 2006], [Grohs, 2011])

$$\begin{aligned} \mathrm{WF}(f)^c \quad = \quad & \Big\{ (t_0, s_0) \in \mathbb{R}^2 \times [-1,1] : \text{for } (t,s) \text{ in neighborhood } U \text{ of } (t_0,s_0): \\ & |\mathcal{SH}_\psi f(a,s,t)| = \mathcal{O}(a^k) \text{ as } a \longrightarrow 0, \forall k \in \mathbb{N}, \text{ unif. over } U \Big\} \end{aligned}$$

Shearlets can Separate the Visible and Invisible Part

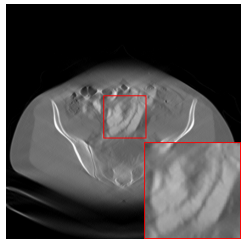


Our Approach "Learn the Invisible (LtI)"

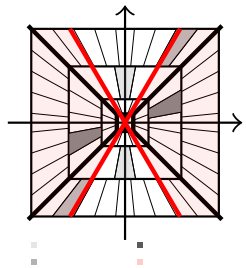
(Bubba, Kutyniok, Lassas, März, Samek, Siltanen, Srinivan; 2018)

Step 1: *Reconstruct the visible*

$$f^* := \operatorname{argmin}_{f \geq 0} \| \mathcal{R}_\phi f - g \|_2^2 + \| \mathrm{SH}_\psi(f) \|_{1,w}$$



- Best available classical solution (little artifacts, denoised)
- Access "wavefront set" via sparsity prior on shearlets:
  - For $(j,k,l) \in \mathcal{I}_{\mathtt{inv}}$: $\mathrm{SH}_\psi(f^*)_{(j,k,l)} \approx 0$
  - For $(j,k,l) \in \mathcal{I}_{\mathtt{vis}}$: $\mathrm{SH}_\psi(f^*)_{(j,k,l)}$ reliable and near perfect



Step 2: *Learn the invisible*

$$\mathcal{NN}_\theta : \mathrm{SH}_\psi(f^*)_{\mathcal{I}_{\mathtt{vis}}} \longrightarrow F \left(\overset{!}{\approx} \mathrm{SH}_\psi(f_{\mathtt{gt}})_{\mathcal{I}_{\mathtt{inv}}} \right)$$

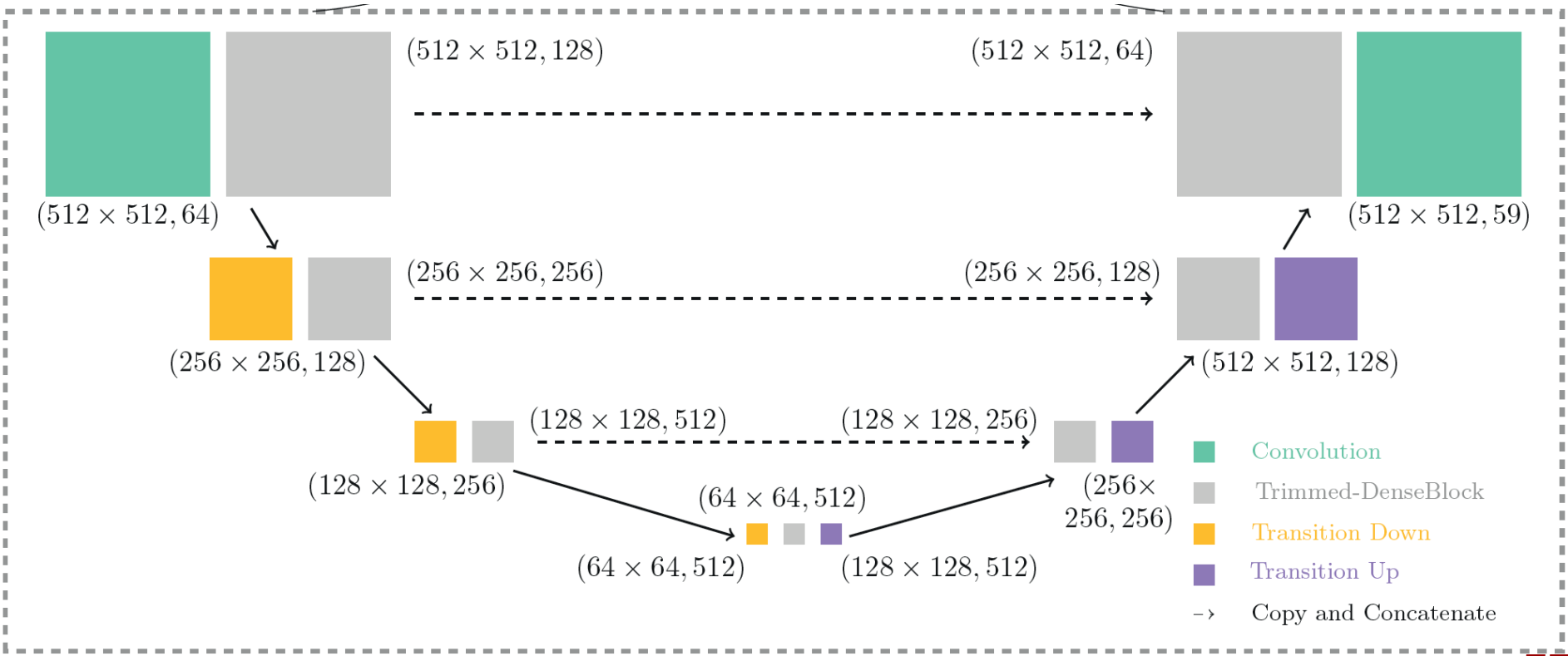
Step 3: *Combine*

$$f_{\mathtt{LtI}} = \mathrm{SH}_\psi^T \left(\mathrm{SH}_\psi(f^*)_{\mathcal{I}_{\mathtt{vis}}} + F \right)$$

Our Approach – Step 2: PhantomNet

U-Net-like CNN architecture $\mathcal{NN}_\theta$ (40 layers) that is trained by minimizing:

$$\min_\theta \frac{1}{N} \sum_{j=1}^{N} \|\mathcal{NN}_\theta(\mathsf{SH}(f_j^*)) - \mathsf{SH}(f_j^{\mathtt{gt}})_{\mathcal{I}_{\mathtt{inv}}}\|_{w,2}^2.$$



Learning the Invisible

Model Based & Data Driven: Only learn what needs to be learned!

Advantages over Pure Data Based Approach:

- Interpretation of what the CNN does ($\rightsquigarrow$ 3D inpainting)
- Reliability by learning only what is *not visible* in the data
- Better performance due to better input
- The neural network does not process entire image, leading to...
  - ...less blurring by U-net
  - ...fewer unwanted artifacts
- Better generalization

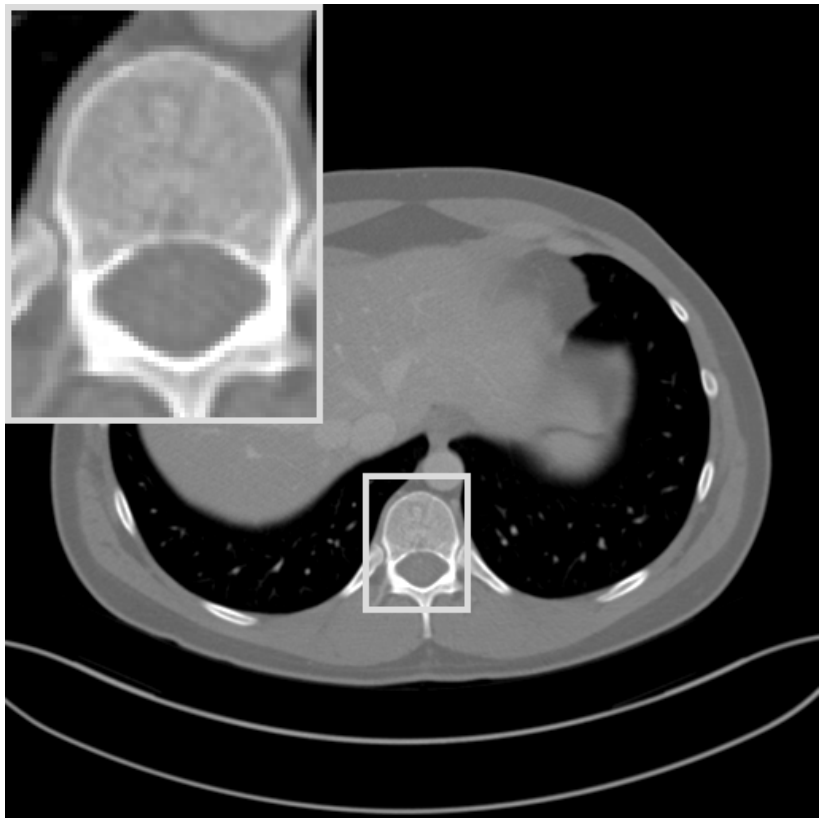
Setup

Experimental Scenarios:

- Mayo Clinic[1]: human abdomen scans provided by the Mayo Clinic for the AAPM Low-Dose CT Grand Challenge.
  - 10 patients (2378 slices of size 512×512 with thickness 3mm)
  - 9 patients for training (2134 slices) and 1 patient for testing (244 slices)
  - simulated noisy *fanbeam* measurements for 60° missing wedge
- Lotus Root: real data measured with the μCT in Helsinki
  - generalization test of our method (training is on Mayo data!)
  - 30° missing wedge
- ...

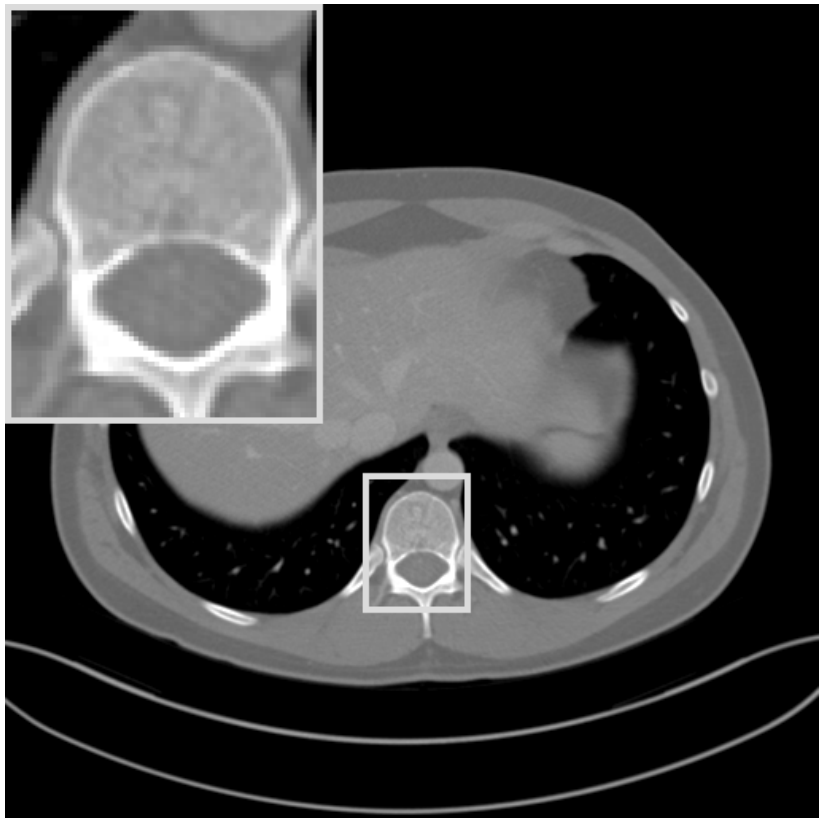
[1]We would like to thank Dr. Cynthia McCollough, the Mayo Clinic, the American Association of Physicists in Medicine (AAPM), and grant EB01705 and EB01785 from the National Institute of Biomedical Imaging and Bioengineering for providing the Low-Dose CT Grand Challenge data set.

Evaluation on Test Patient

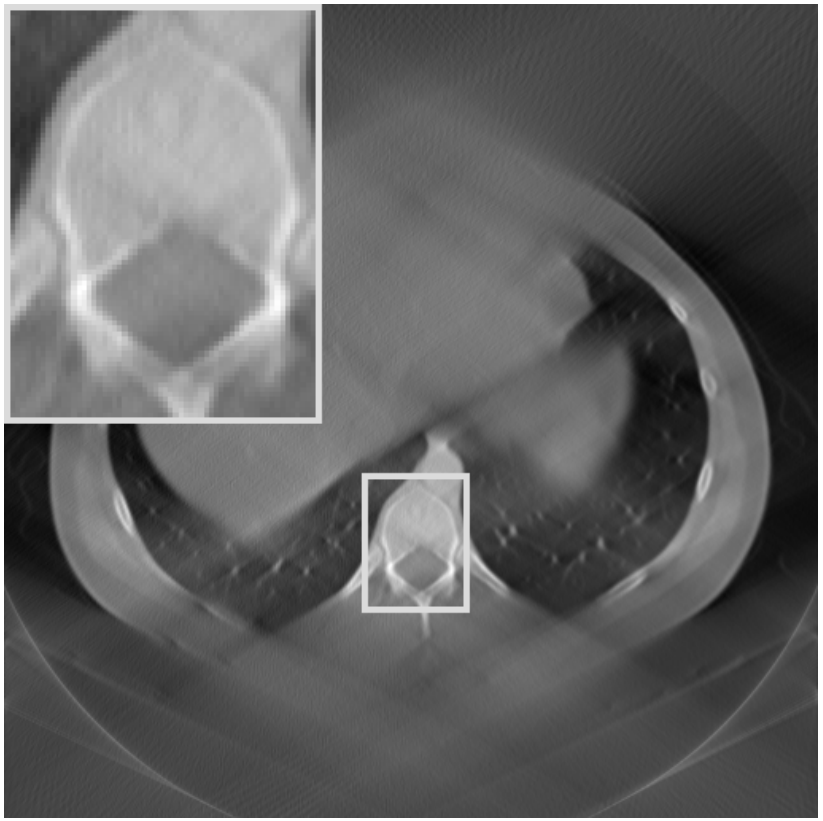


f_{gt}

Evaluation on Test Patient

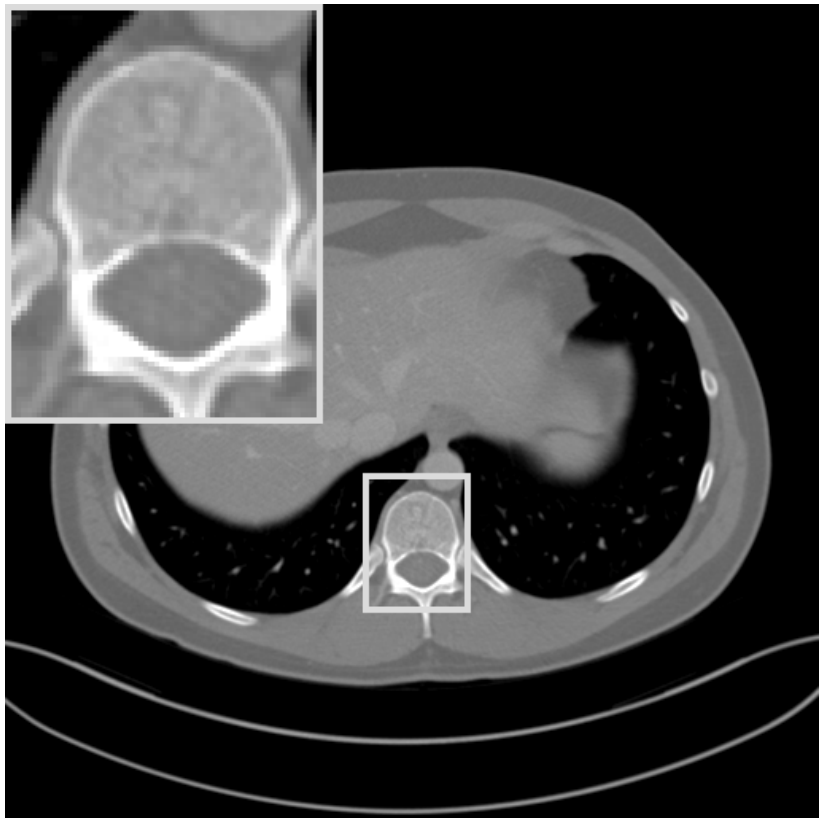


f_{gt}

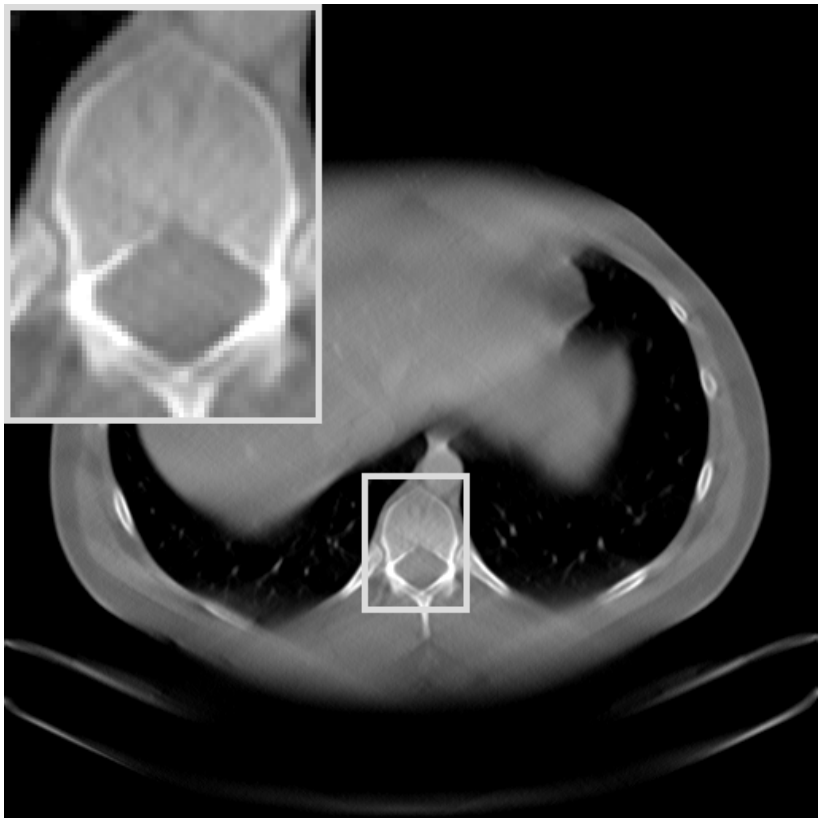


f_{FBP}: RE $= 0.50$, HaarPSI$=0.35$

Evaluation on Test Patient

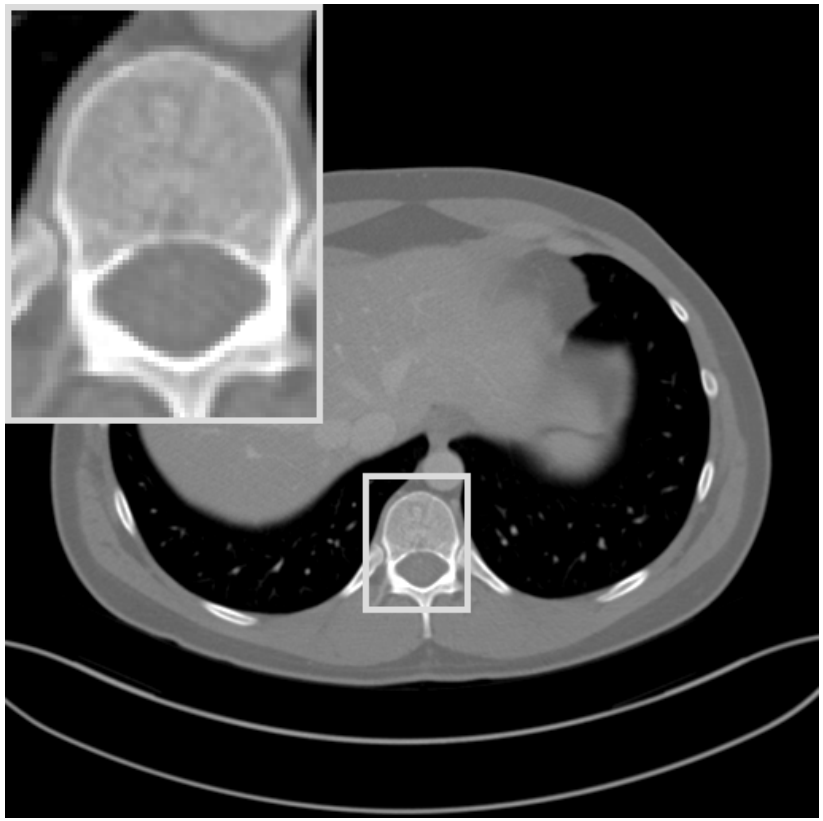


f_{gt}

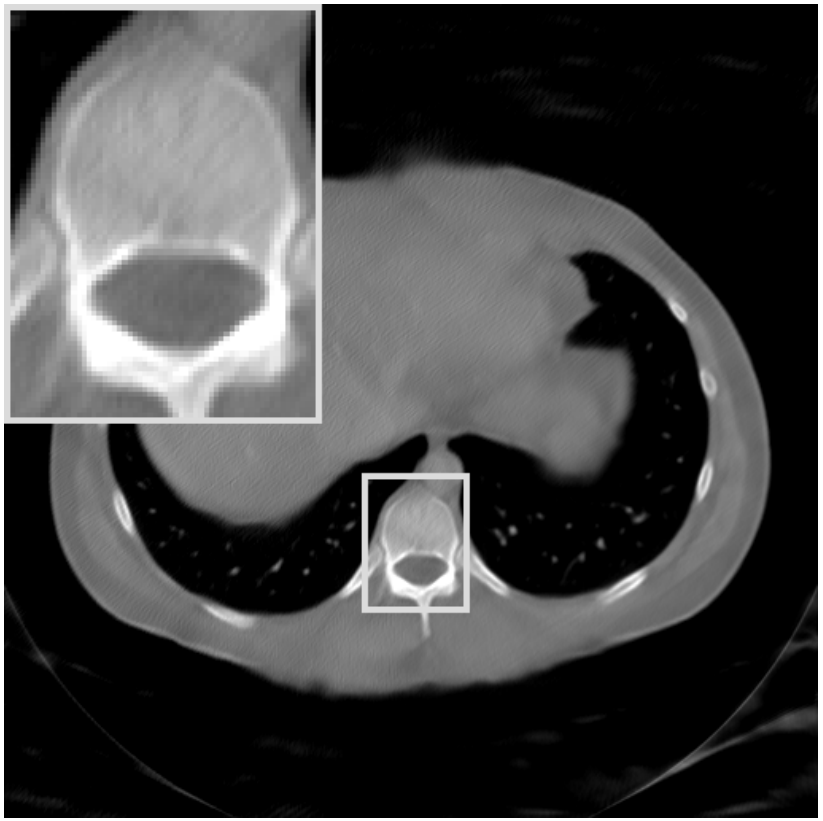


f^*: RE $= 0.19$, HaarPSI$=0.43$

Evaluation on Test Patient

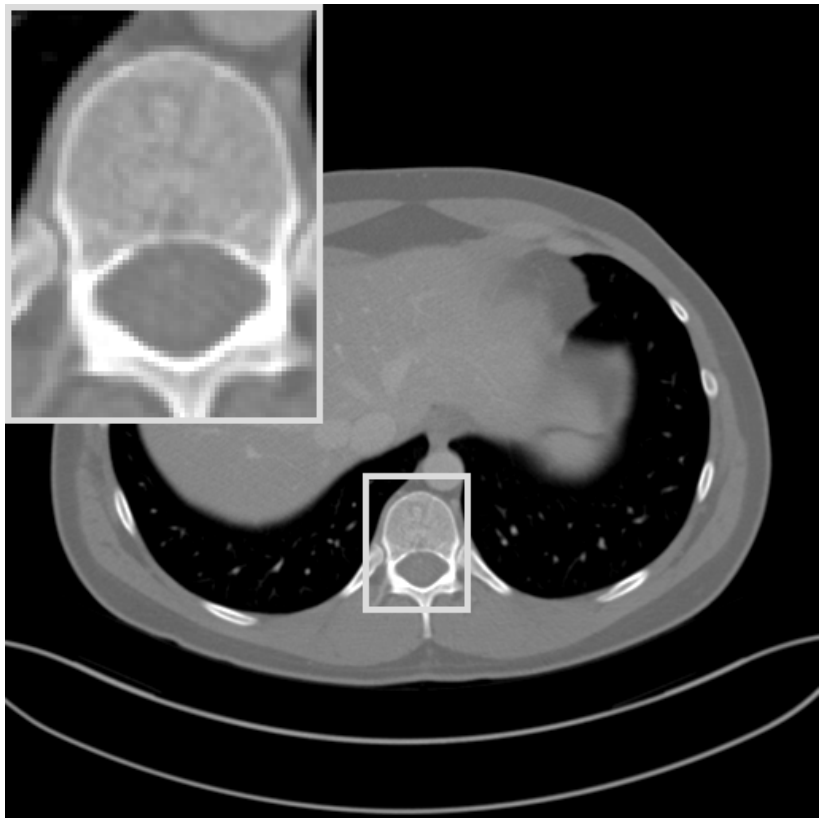


f_{gt}



$f_{[\text{Gu \& Ye, 2017}]}$: RE = 0.22, HaarPSI=0.40

Evaluation on Test Patient



f_{gt}

$f_{\texttt{LtI}}$: RE = 0.09, HaarPSI=0.76

Average over Test Patient

Method	RE	PSNR	SSIM	HaarPSI
$f_{\mathtt{FBP}}$	0.47	17.16	0.40	0.32
$f_{\mathtt{TV}}$	0.18	25.88	0.85	0.37
f^*	0.17	26.34	0.85	0.40
$f_{\text{[Gu \& Ye, 2017]}}$	0.25	23.06	0.61	0.34
$\mathcal{NN}_\theta(f_{\mathtt{FBP}})$	0.15	27.40	0.78	0.52
$\mathcal{NN}_\theta(\mathsf{SH}(f_{\mathtt{FBP}}))$	0.16	26.80	0.74	0.52
$f_{\mathtt{LtI}}$	**0.08**	**32.77**	**0.93**	**0.73**

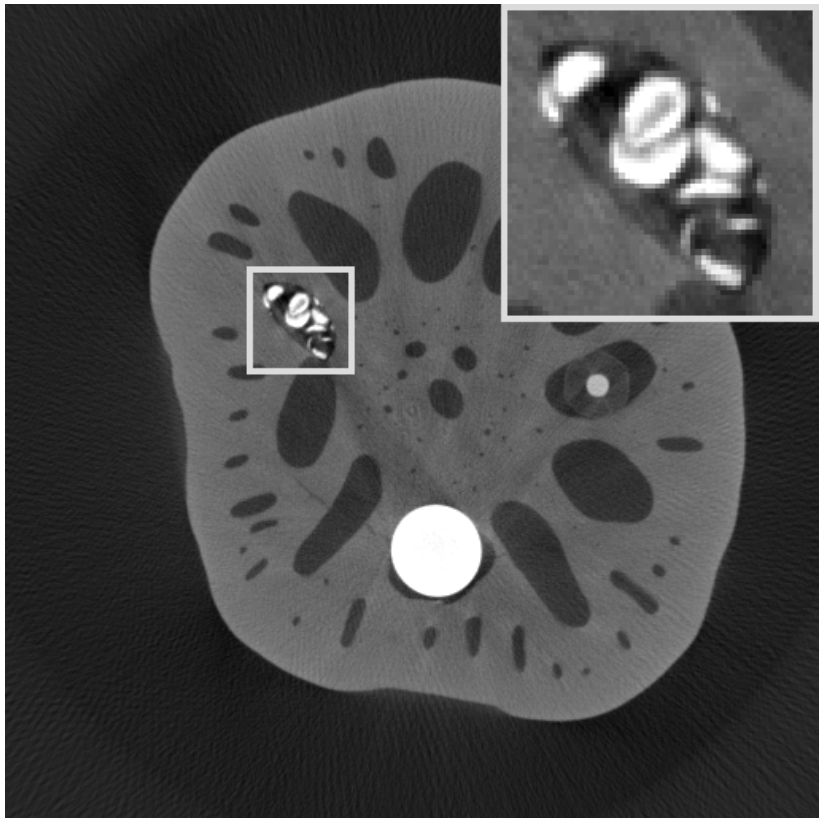
HaarPSI (Reisenhofer, Bosse, Kutyniok, and Wiegand; 2018)

Advantages over (MS-)SSIM, FSIM, PSNR, GSM, VIF, etc.:

- Achieves higher correlations with human opinion scores.
- Can be computed very efficiently and significantly faster.

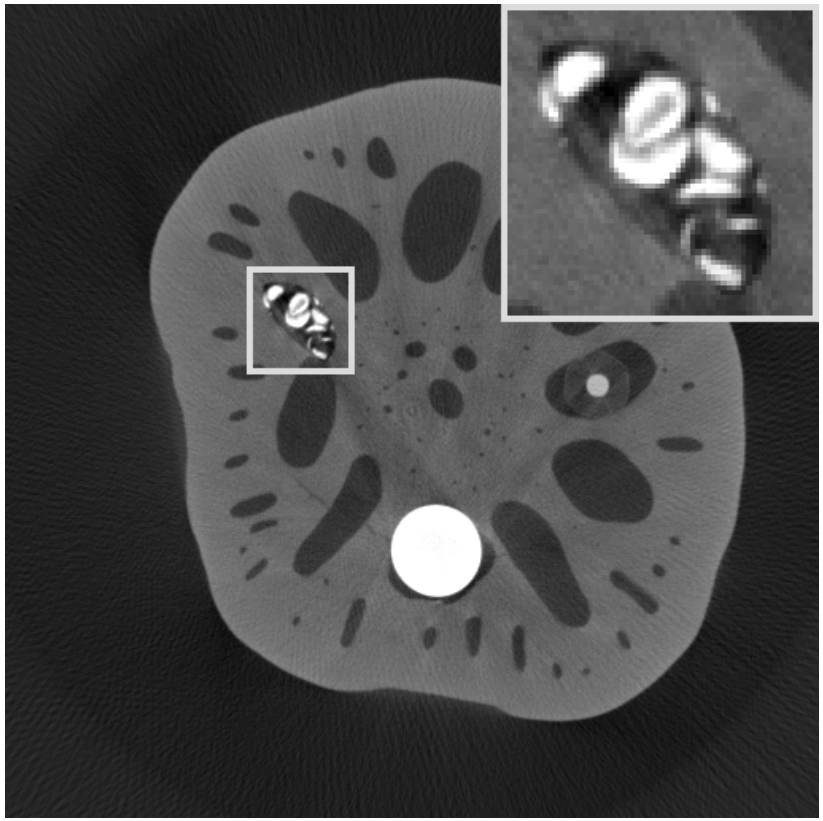
www.haarpsi.org

Generalization to Lotus Root

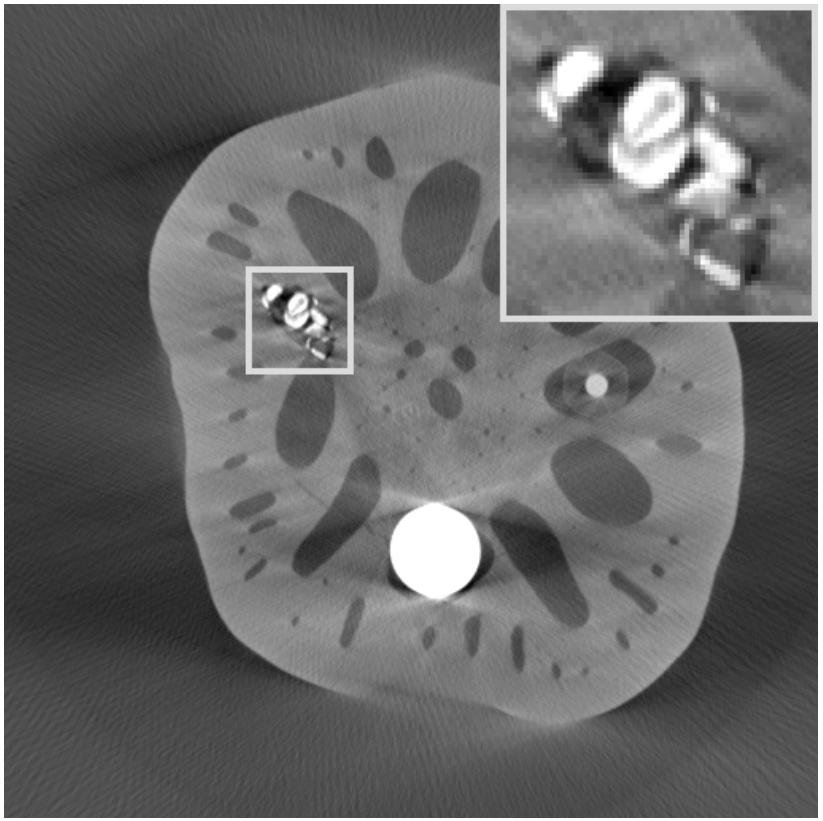


f_{gt}

Generalization to Lotus Root

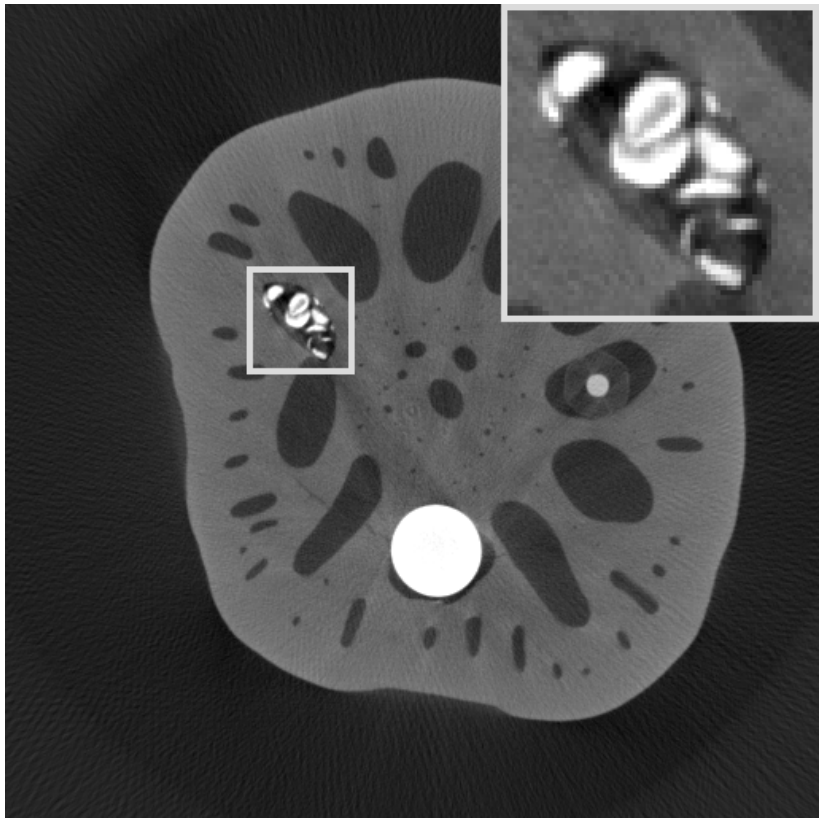


$f_{\mathtt{gt}}$

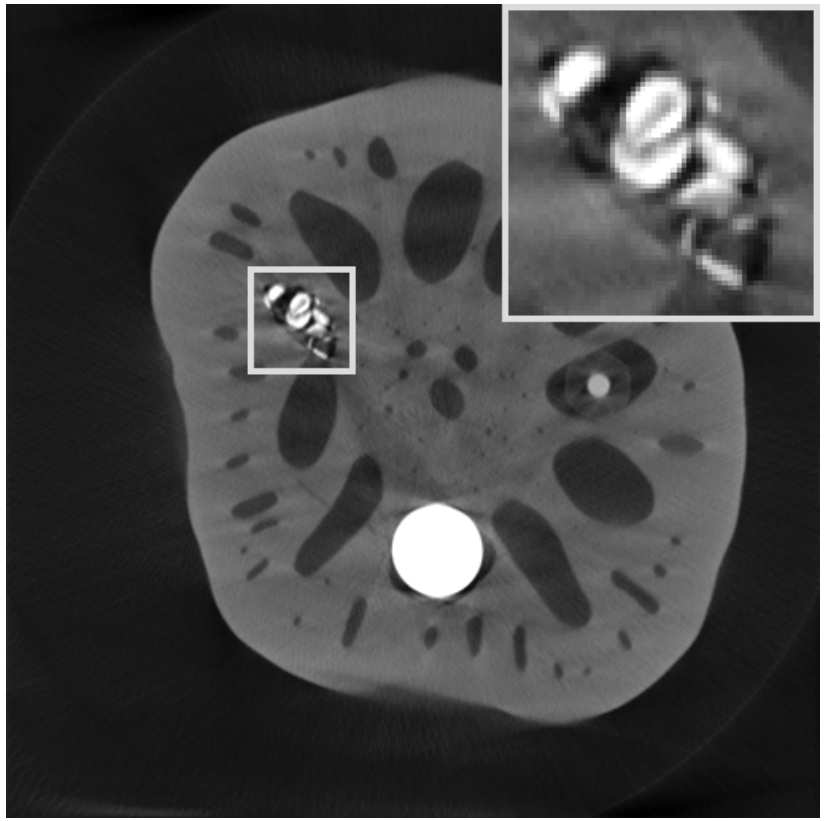


$f_{\mathtt{FBP}}$: RE $= 0.31$, HaarPSI$=0.61$

Generalization to Lotus Root

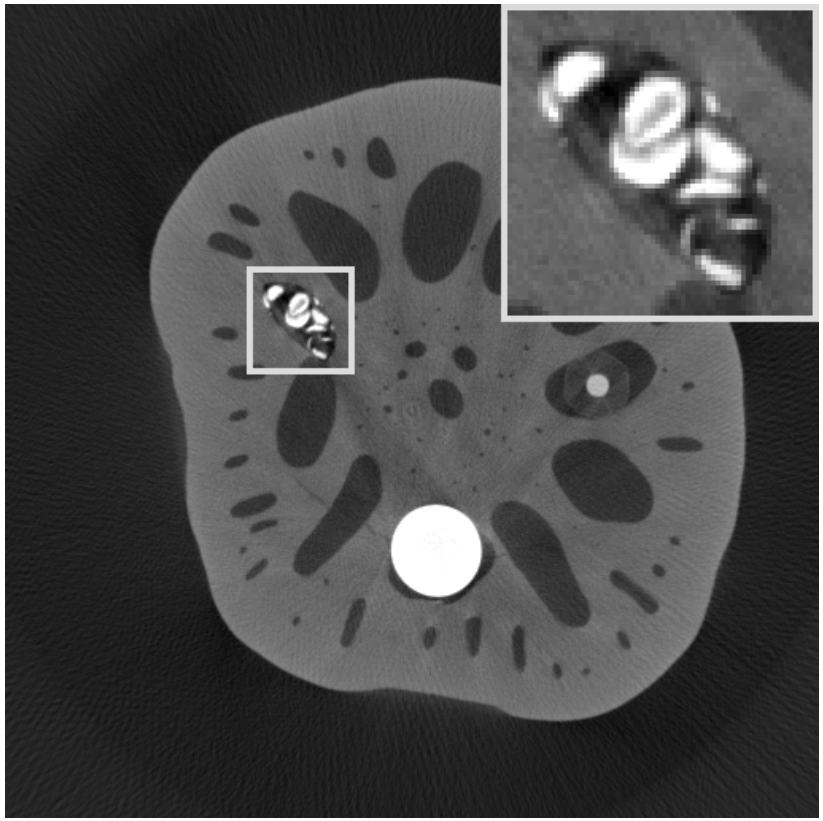


f_{gt}

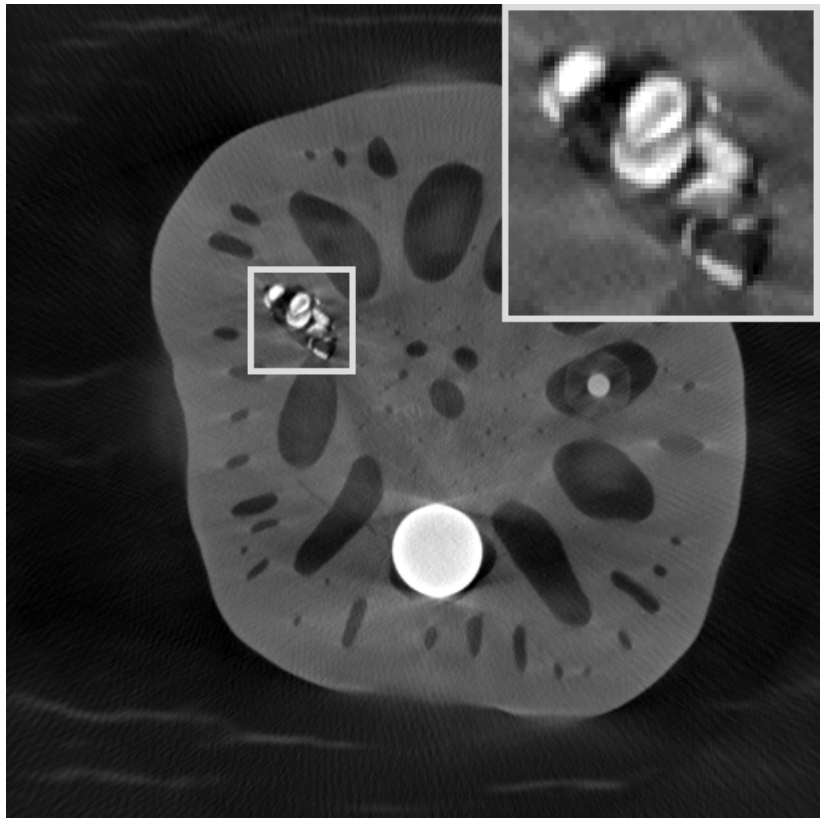


f^*: RE $= 0.11$, HaarPSI$=0.75$

Generalization to Lotus Root

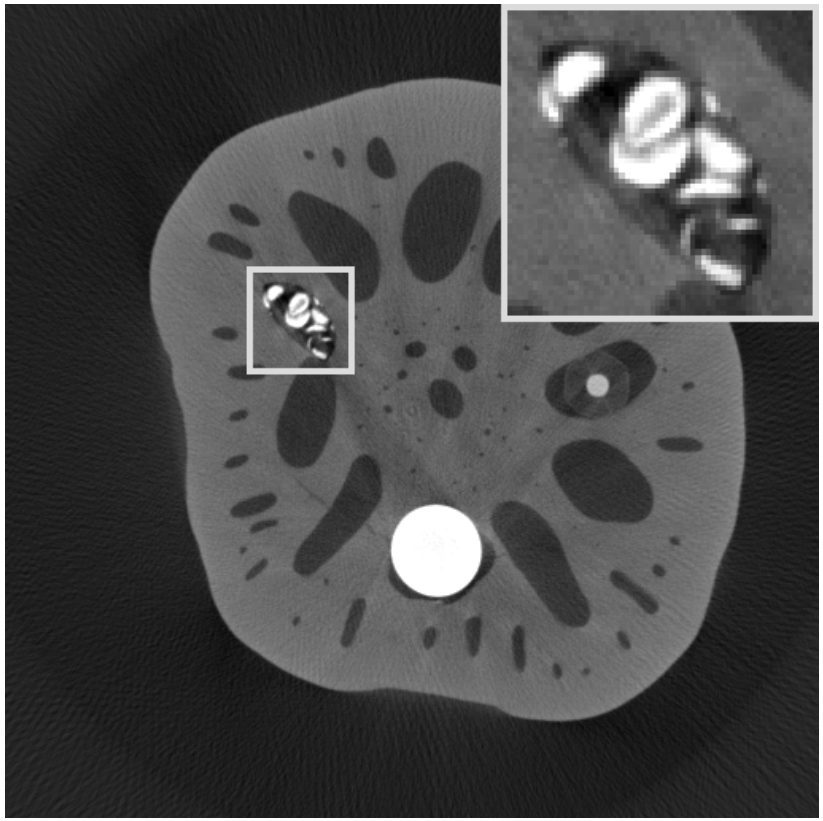


f_{gt}

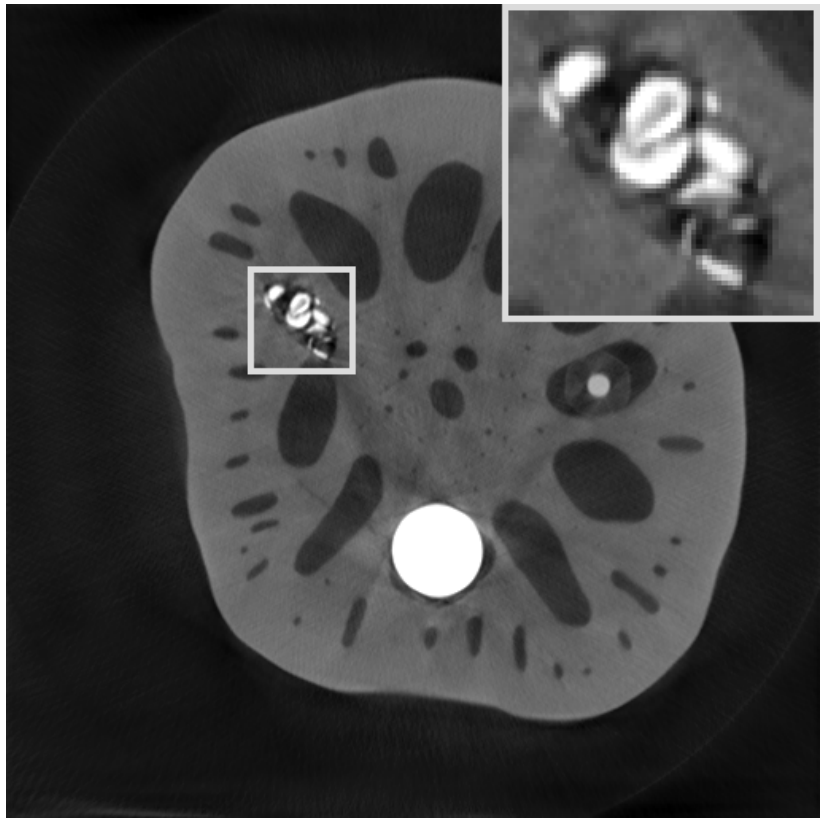


$f_{[\text{Gu \& Ye, 2017}]}$: RE = 0.25, HaarPSI=0.62

Generalization to Lotus Root



f_{gt}



$f_{\texttt{LtI}}$: RE = 0.11, HaarPSI=0.83

Conclusions

What to take Home...?

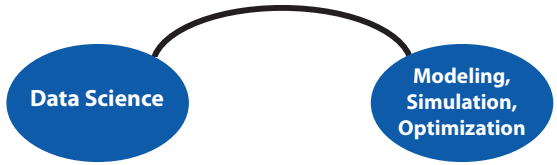
Model-Based Side:

- Traditional (differential-equation based) methodologies *very well understood*.
- Methods based on *mathematical models* today often *reach a barrier*.

Data-Based Side:

- Novel powerful, pure data-based methodologies.
- Deep learning is currently entering numerous applications.
- A *theoretical foundation* is still largely *missing*.

Combining Both Sides:



- *Optimal balancing* of data-adaptiveness and differential equation-based modeling!
- Limited Angle CT: Learn only the *invisible parts* with a *deep neural network*.

THANK YOU!

References available at:

www.math.tu-berlin.de/∼kutyniok

Code available at:

www.ShearLab.org

Related Books:

- G. Kutyniok and D. Labate
  Shearlets: Multiscale Analysis for Multivariate Data
  Birkhäuser-Springer, 2012.

- P. Grohs and G. Kutyniok
  Theory of Deep Learning
  Cambridge University Press (in preparation)