FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
FACULTY OF ENGINEERING

An Introduction to Deep Learning Computed Tomography and Known Operator Learning

Andreas Maier

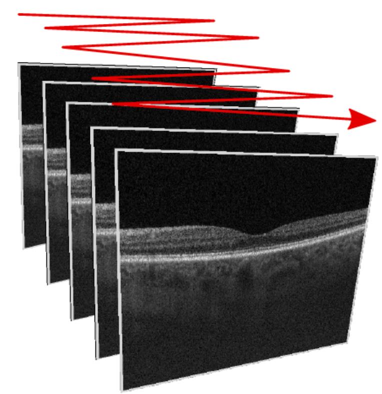
Lehrstuhl für Mustererkennung (Informatik 5),
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

European Research Council

Known Operator Learning

- Introduction
- Current State-of-the-art in Deep Learning
- Prior Operators in Deep Networks
- Future Work

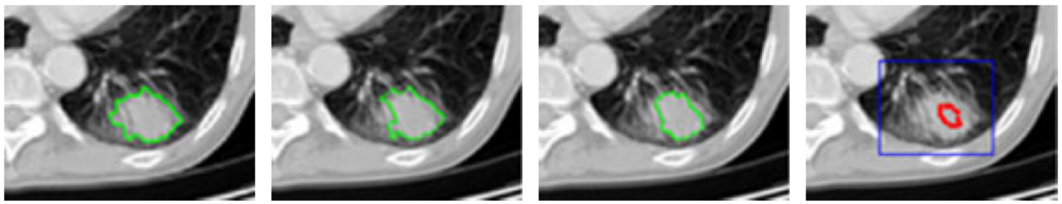
Work at the Lab - Examples

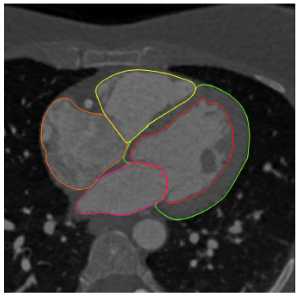


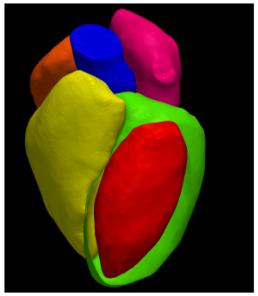
Known Operator Learning

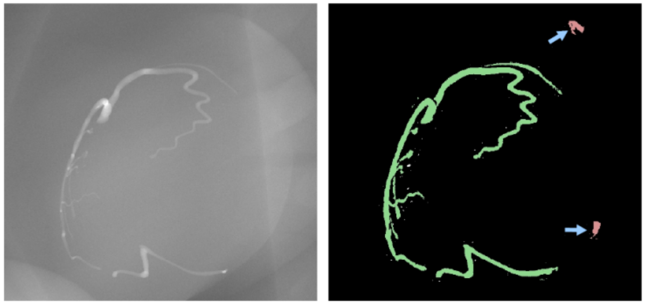
- Introduction
- Current State-of-the-art in Deep Learning
- Prior Operators in Deep Networks
- Future Work

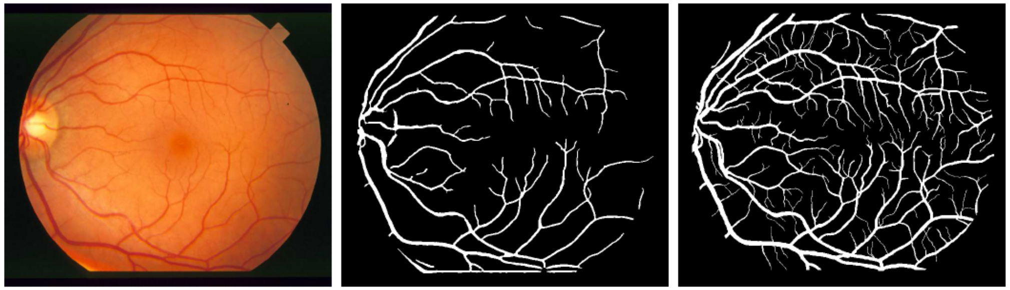
Work at the Lab - Examples

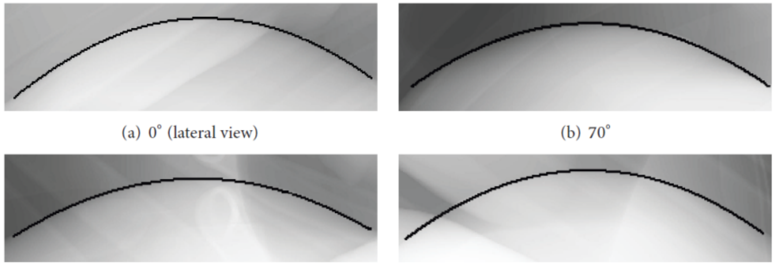










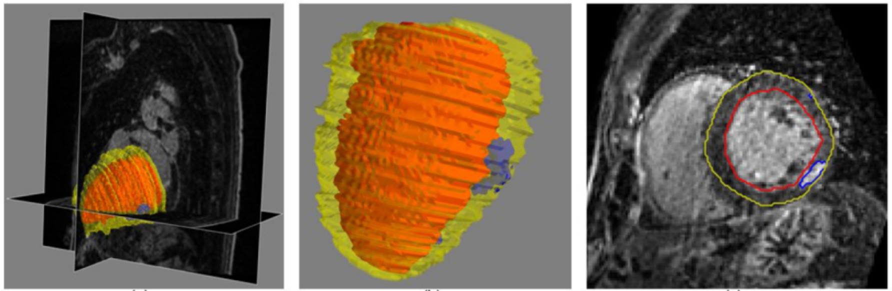


(a) 0° (lateral view)

(b) 70°

(c) 140°

(d) 200°



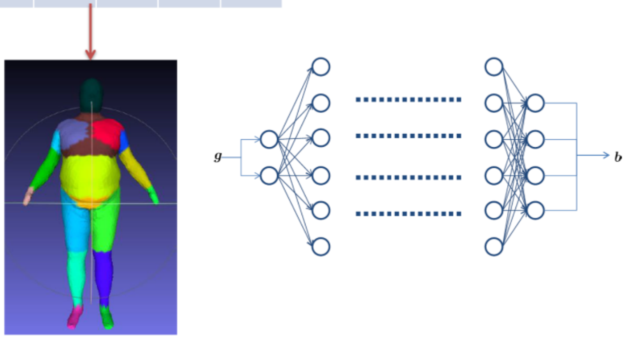
(a)

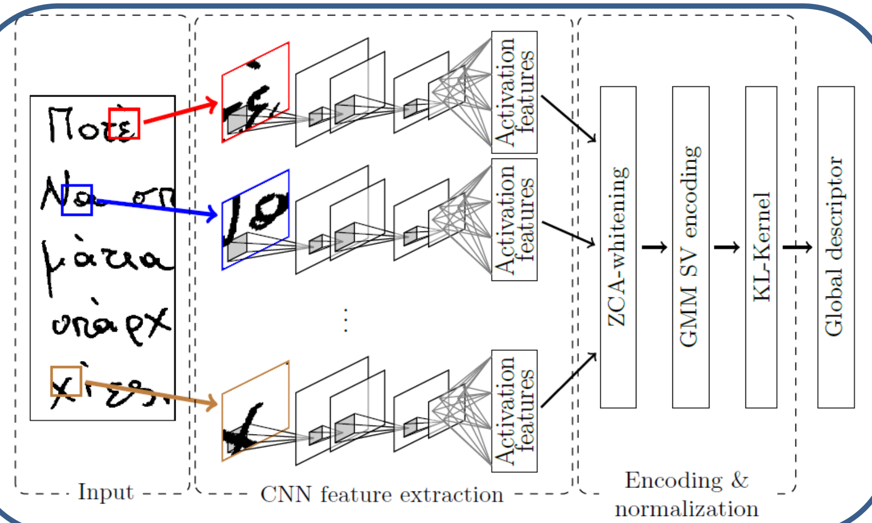
(b)

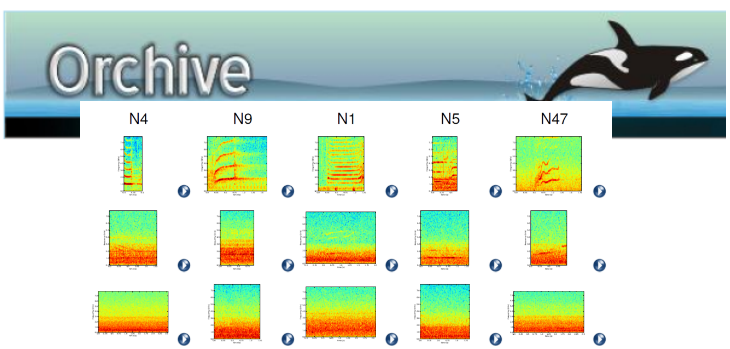
(c)

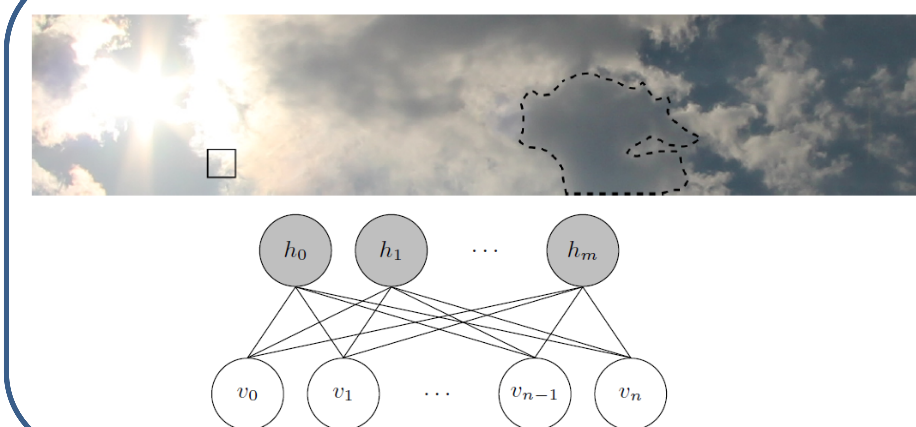
Work at the Lab – Examples Deep Learning

G	H	W	J	P/W	P/I
M	187	98	48	36	33





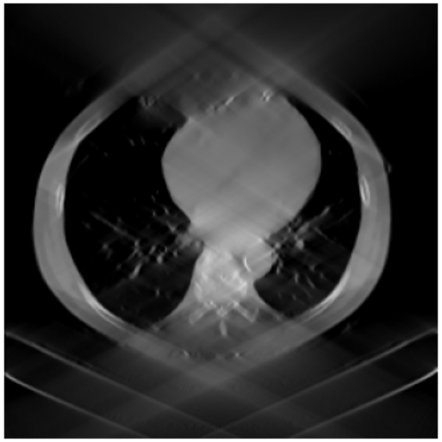


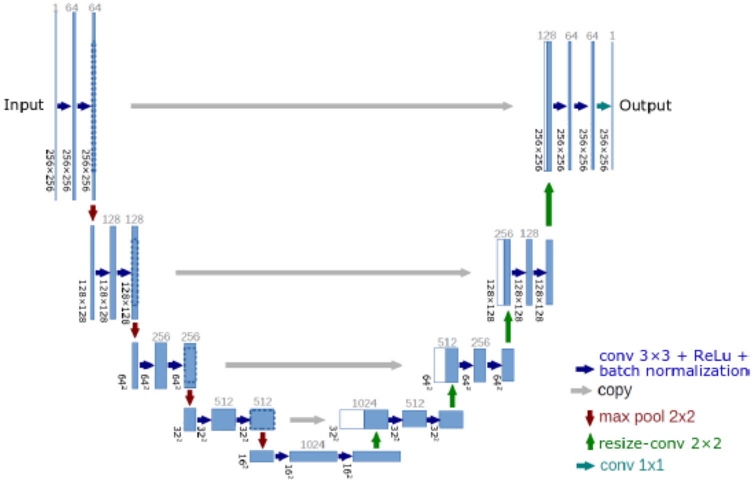


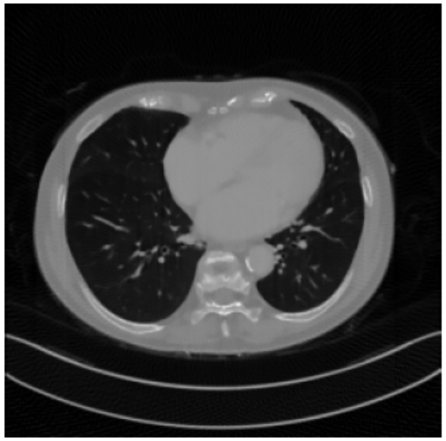
Known Operator Learning

- Introduction
- Current State-of-the-art in Deep Learning
- Prior Operators in Deep Networks
- Future Work

DL – Deep Learning Image Reconstruction?

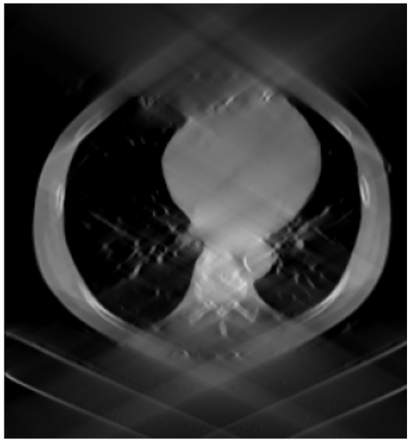


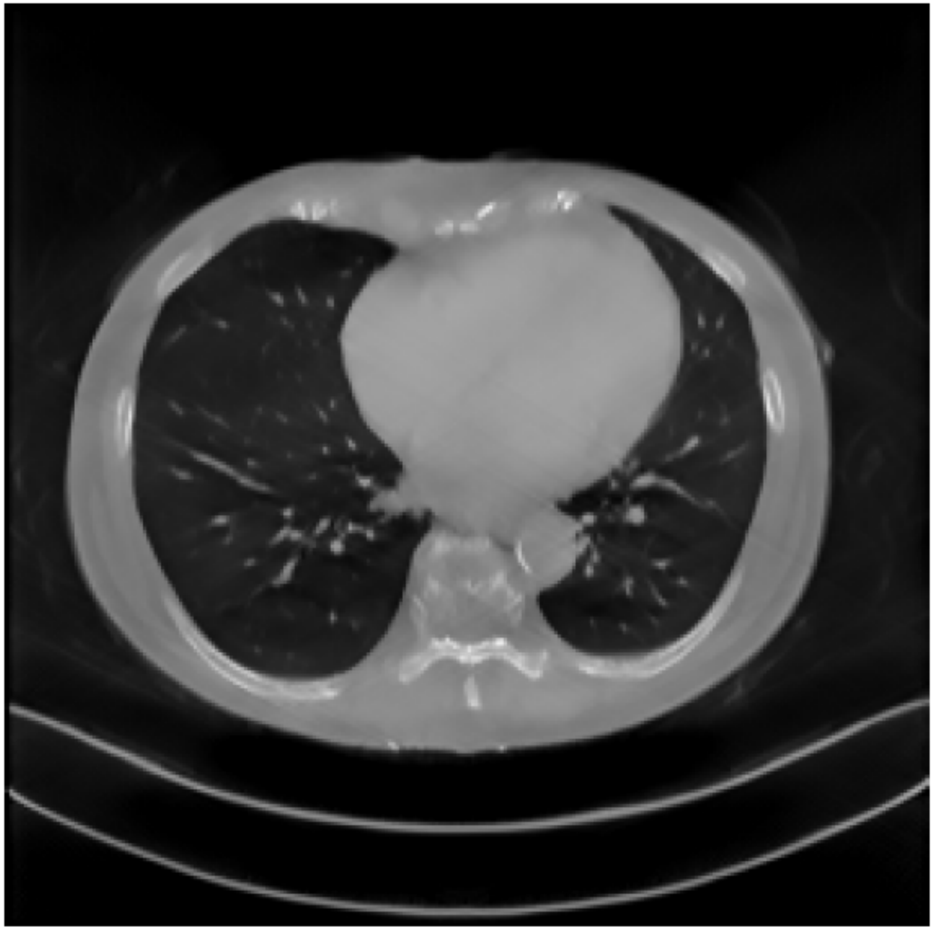


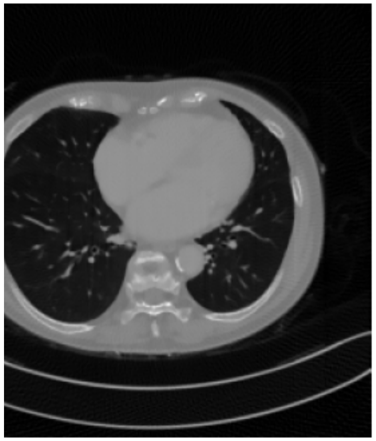


[4] Yixing Huang et al. Some Investigations on Robustness of Deep Learning in Limited Angle Tomography. MICCAI 2018.

DL – Deep Learning Image Reconstruction?

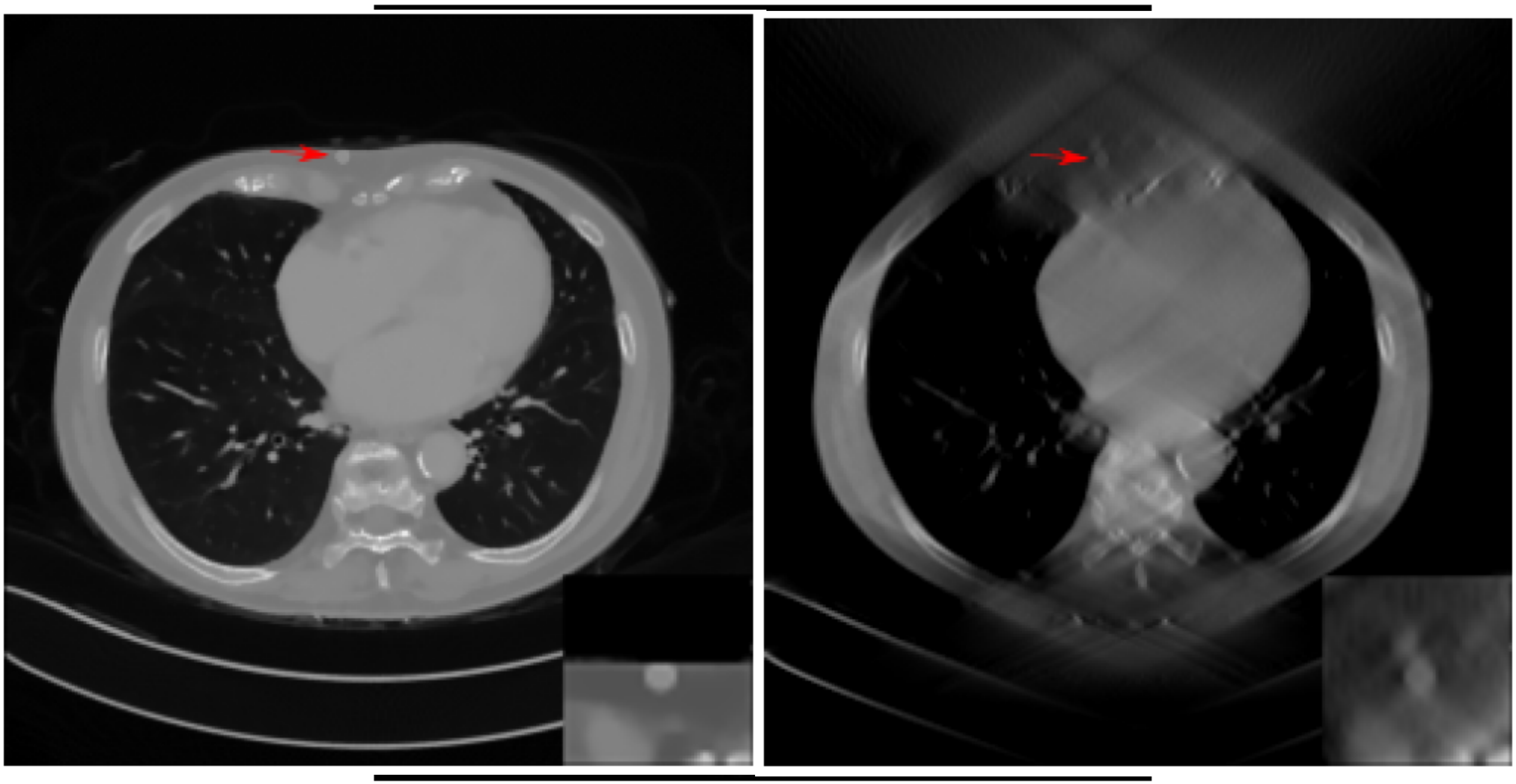






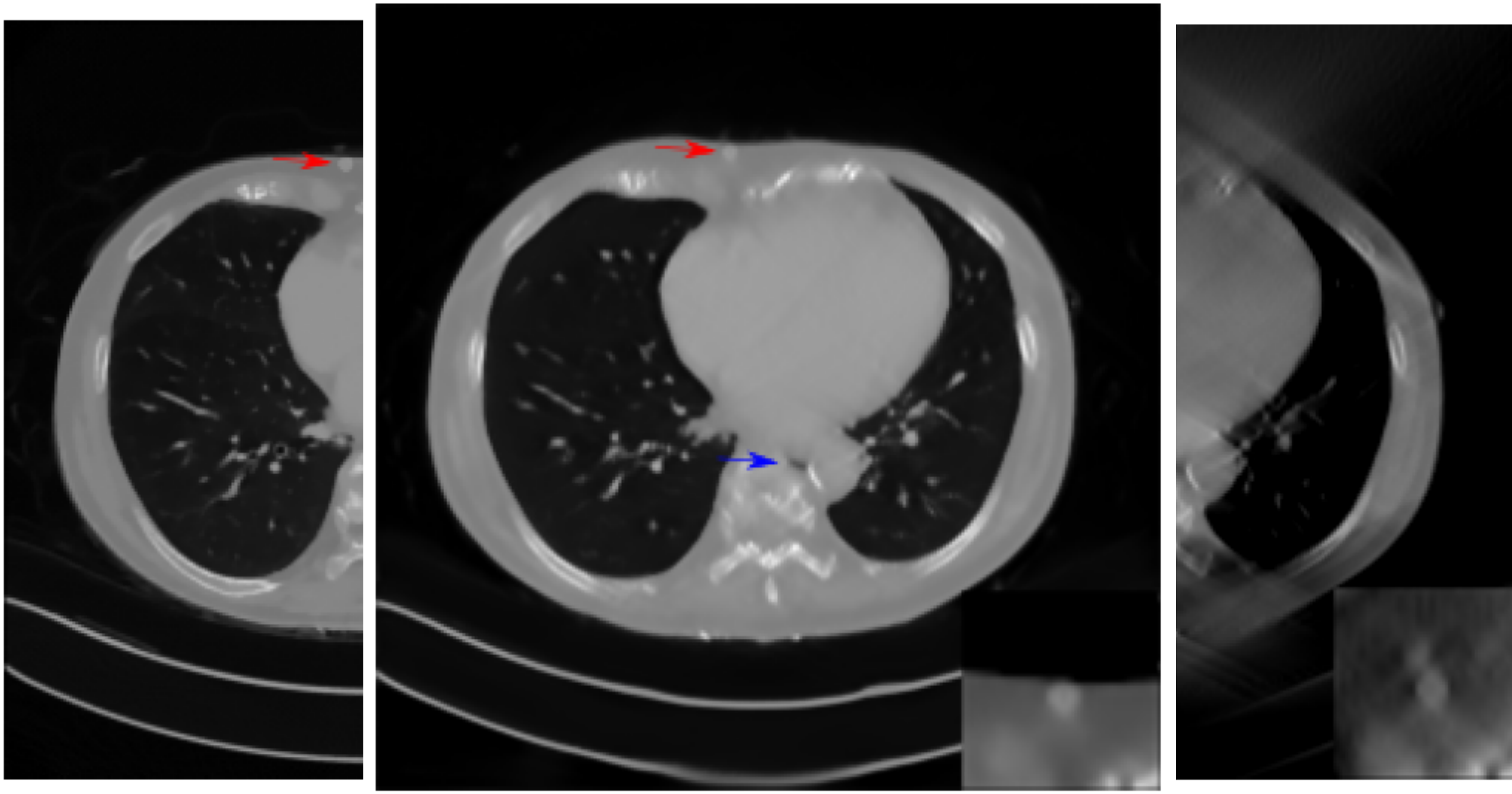
[4] Yixing Huang et al. Some Investigations on Robustness of Deep Learning in Limited Angle Tomography. MICCAI 2018.

DL – Deep Learning Image Reconstruction?



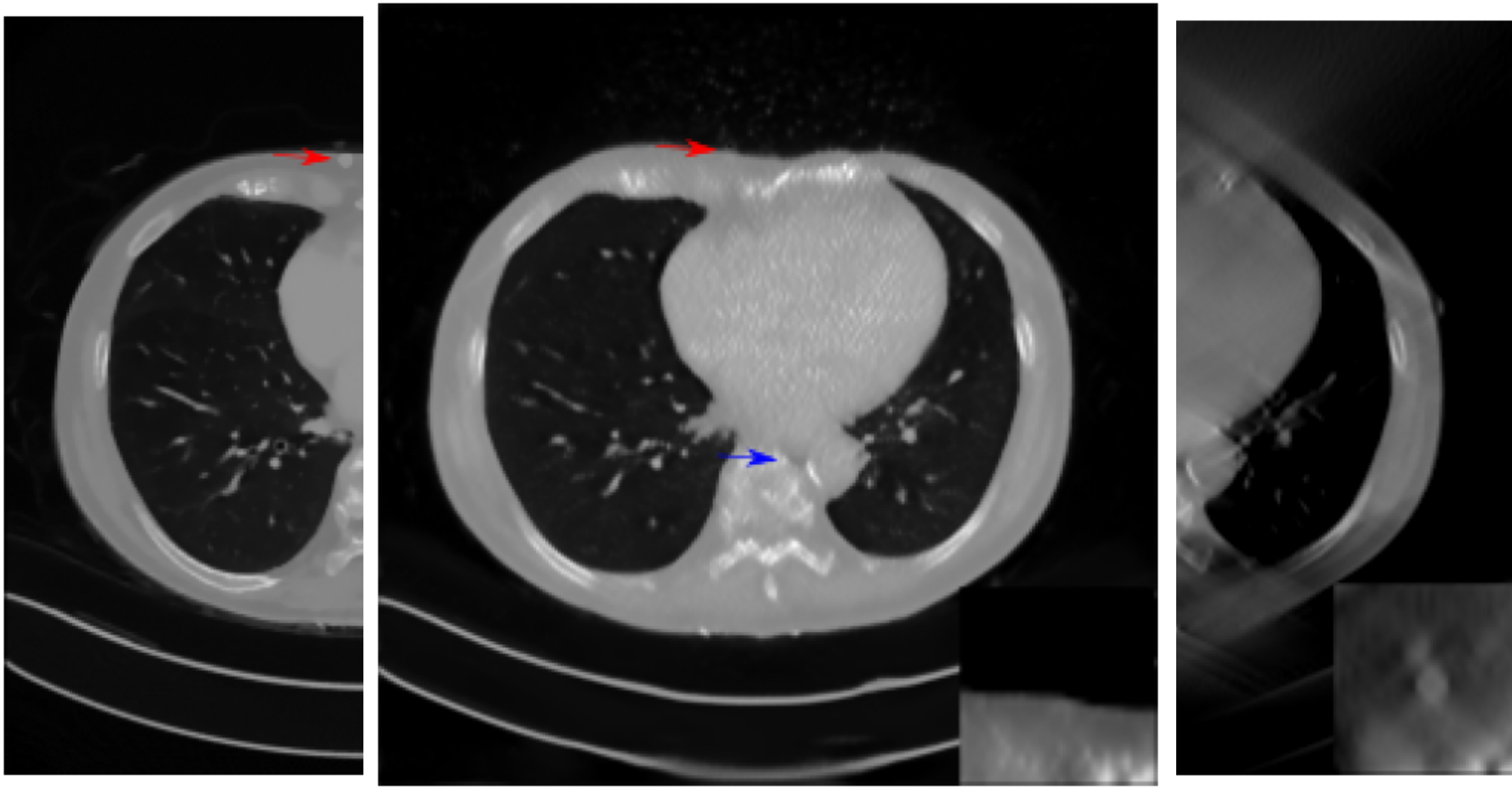
[4] Yixing Huang et al. Some Investigations on Robustness of Deep Learning in Limited Angle Tomography. MICCAI 2018.

DL – Deep Learning Image Reconstruction?



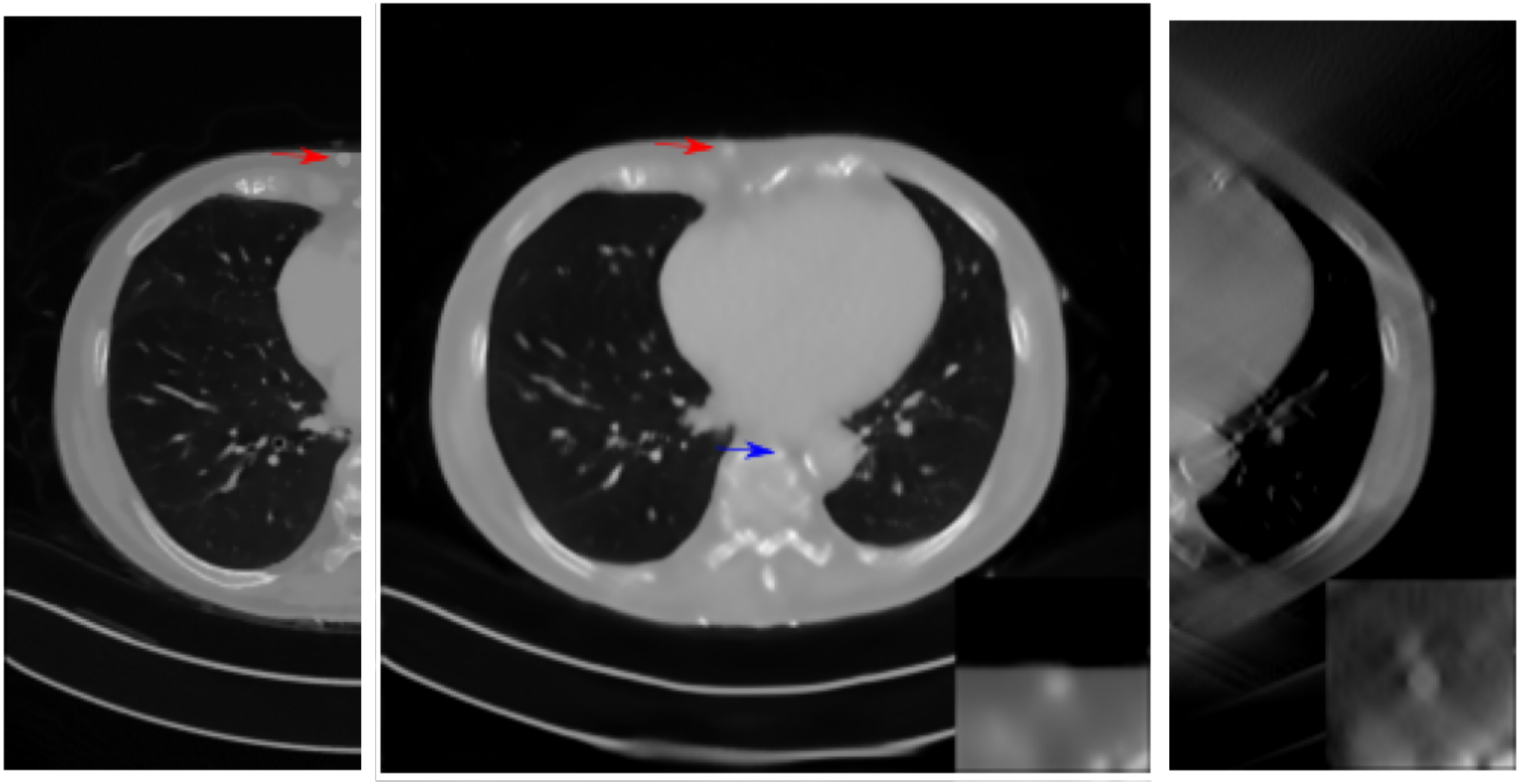
[4] Yixing Huang et al. Some Investigations on Robustness of Deep Learning in Limited Angle Tomography. MICCAI 2018.

DL – Deep Learning Image Reconstruction?



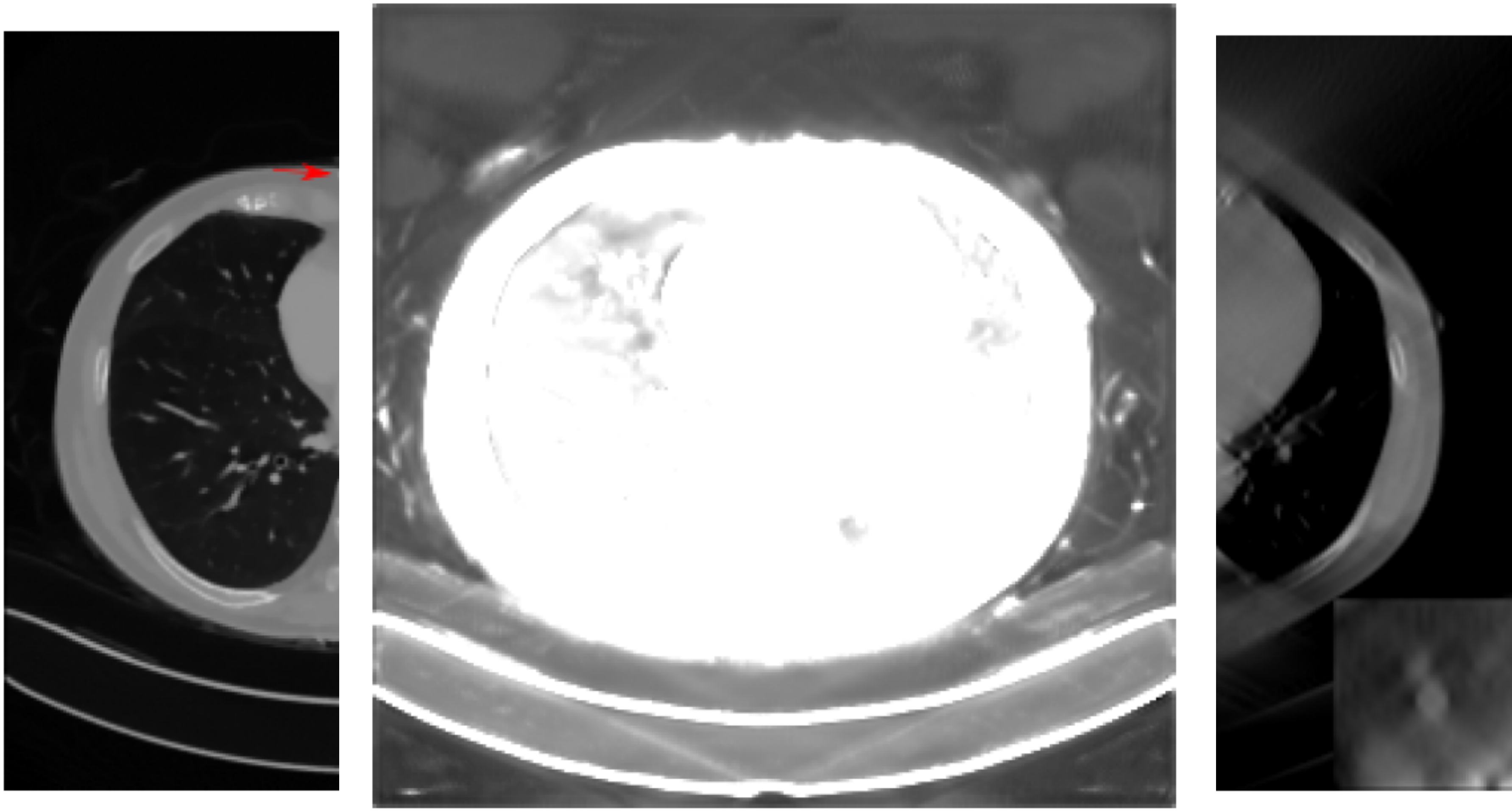
[4] Yixing Huang et al. Some Investigations on Robustness of Deep Learning in Limited Angle Tomography. MICCAI 2018.

DL – Deep Learning Image Reconstruction?



[4] Yixing Huang et al. Some Investigations on Robustness of Deep Learning in Limited Angle Tomography. MICCAI 2018.

DL – Deep Learning Image Reconstruction?



[4] Yixing Huang et al. Some Investigations on Robustness of Deep Learning in Limited Angle Tomography. MICCAI 2018.

Prior Operators in Neural Networks

"Let’s not reinvent the wheel…"

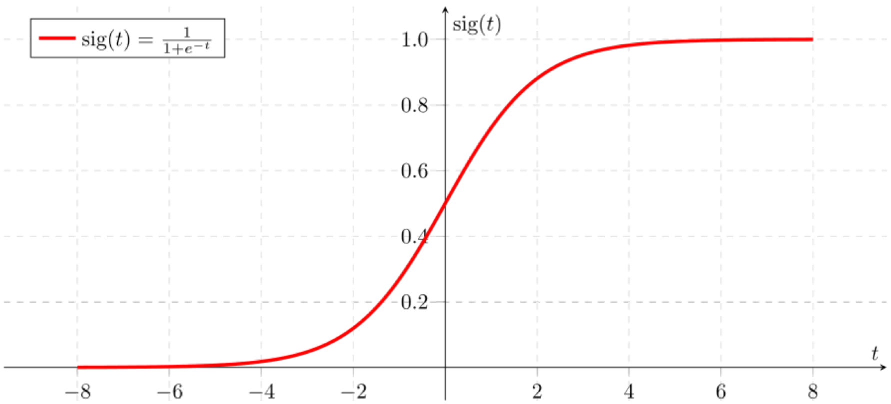
Universal Approximation Theorem

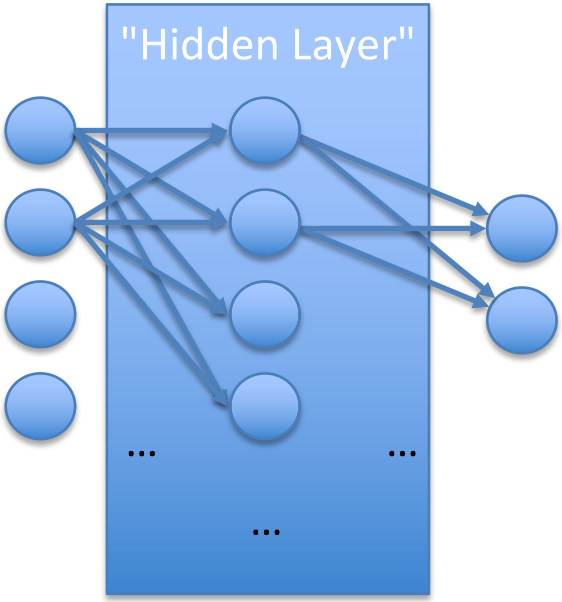
- **Any continuous function can be approximated by Neural Net**

$$u(\mathbf{x}) \approx U(\mathbf{x}) = \sum_i u_i s(\mathbf{w}_i^\top \mathbf{x} + w_{j,0})$$

- **The error is bound by**

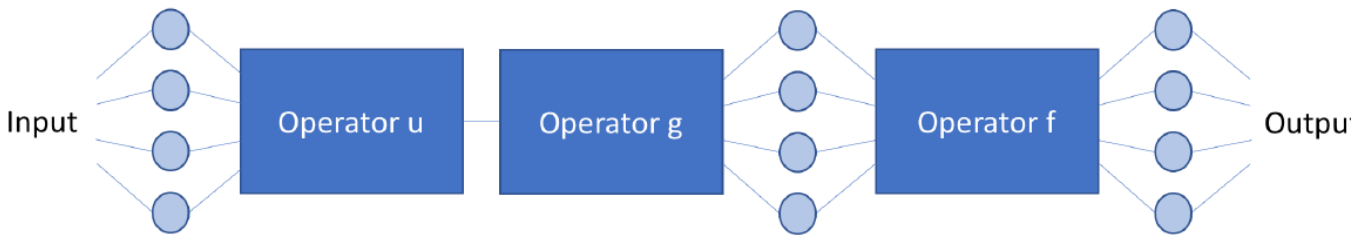
$$|U(\mathbf{x}) - u(\mathbf{x})| \leq \epsilon_u$$





Prior Operators – Precision Learning

- **Consider the case of using known operators in the net**



- **Specifically consider the use of two operators in sequence**

$$f(\mathbf{x}) = g(\mathbf{u}(\mathbf{x}))$$

[5] Andreas Maier et al. Precision Learning: Towards use of known operators in neural networks. ICPR 2018..

Approximation Sequences

- **Sequential operations**

$$f(\mathbf{x}) = g(\mathbf{u}(\mathbf{x}))$$

- **Can be approximated:**

$$\begin{aligned} F_u(\mathbf{x}) &= g(\mathbf{U}(\mathbf{x})) &= f(\mathbf{x}) - e_u \\ F_g(\mathbf{x}) &= G(\mathbf{u}(\mathbf{x})) &= f(\mathbf{x}) - e_g \\ F(\mathbf{x}) &= G(\mathbf{U}(\mathbf{x})) &= f(\mathbf{x}) - e_f \end{aligned}$$

Error of Approximation Sequences

- **Approximation introduces error**

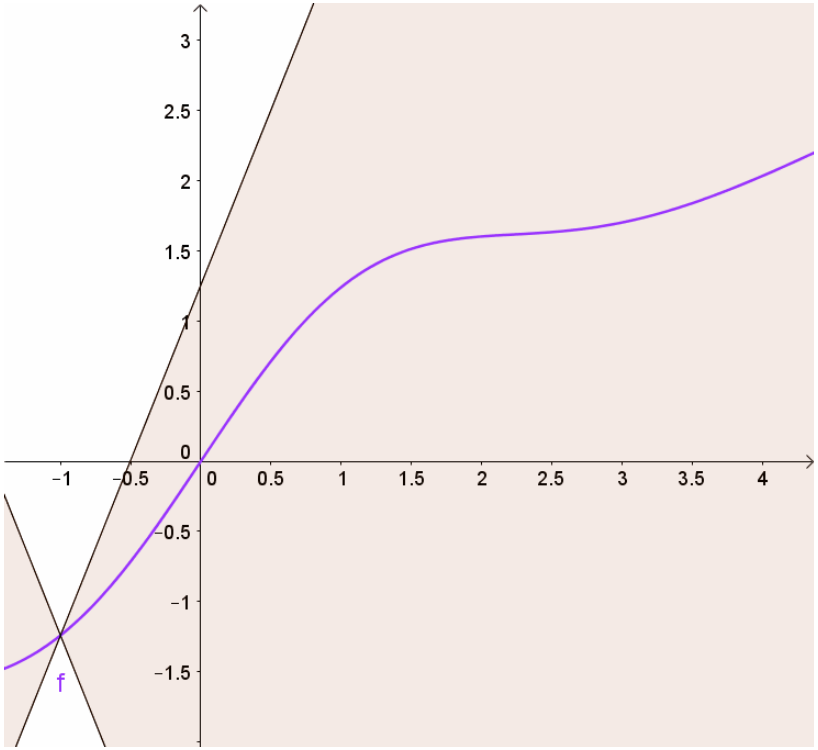
$$
\begin{aligned}
f(\mathbf{x}) &= g(\mathbf{u}(\mathbf{x})) = G(\mathbf{u}(\mathbf{x})) + e_g \\
&= \sum_j g_j s\left(u_j(\mathbf{x})\right) + g_0 + e_g \\
&= \sum_j g_j s\left(U_j(\mathbf{x}) + e_{u_j}\right) + g_0 + e_g
\end{aligned}
$$

- **Can we find bounds on this error?**

Bounds for Sigmoid Functions

- **Sigmoid function satisfies the following upper bound:**

$$s(x+e) \leq s(x) + l_s \cdot |e|$$



Observations on Bounds

- **Bound on Error:**

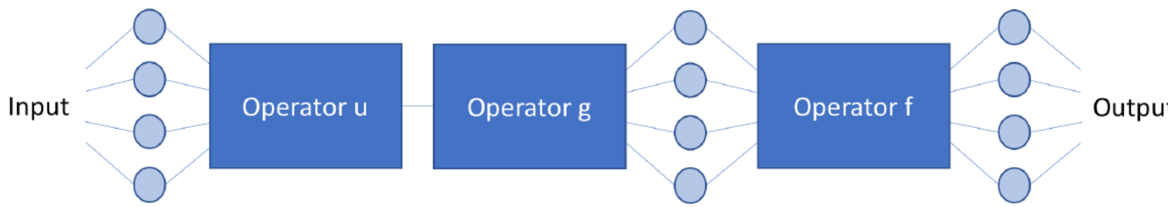
$$|e_f| \leq \sum_j |g_j| \cdot l_s \cdot |e_{u_j}| + \epsilon_g$$

Error U(x) — Error G(x)

- **Observations**
  - **Error in U(x) and G(x) additive**
  - **Error of U(x) amplified by g(x)**
  - **Interpretation as Feature Extractor => Importance of Features**
  - **Requires Lipschitz continuity**

Observations on Bounds

- **Extension to Deep Networks:**



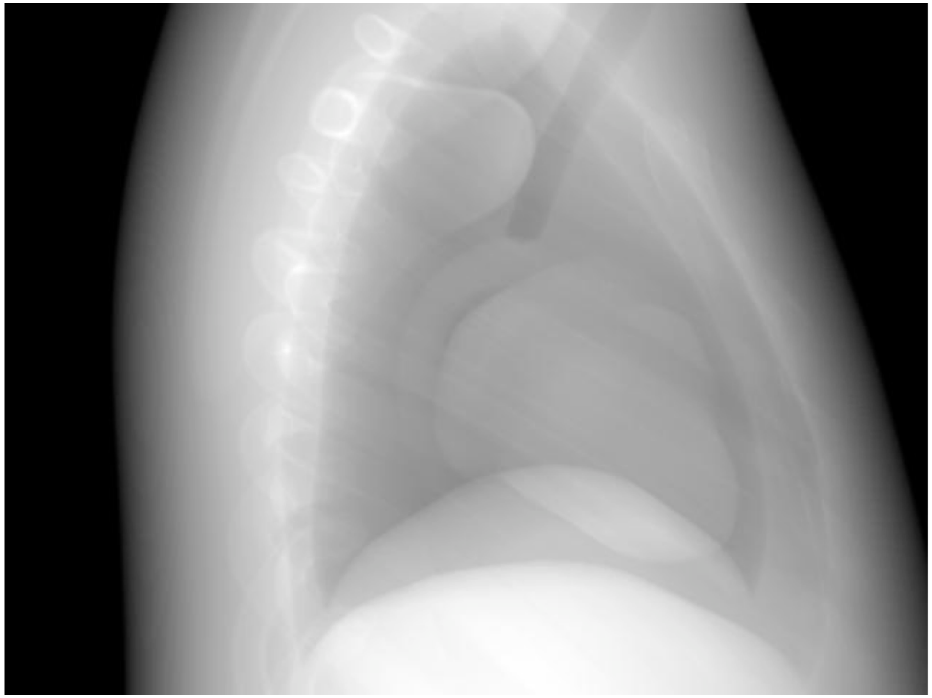
- **Proof by Recursion:**

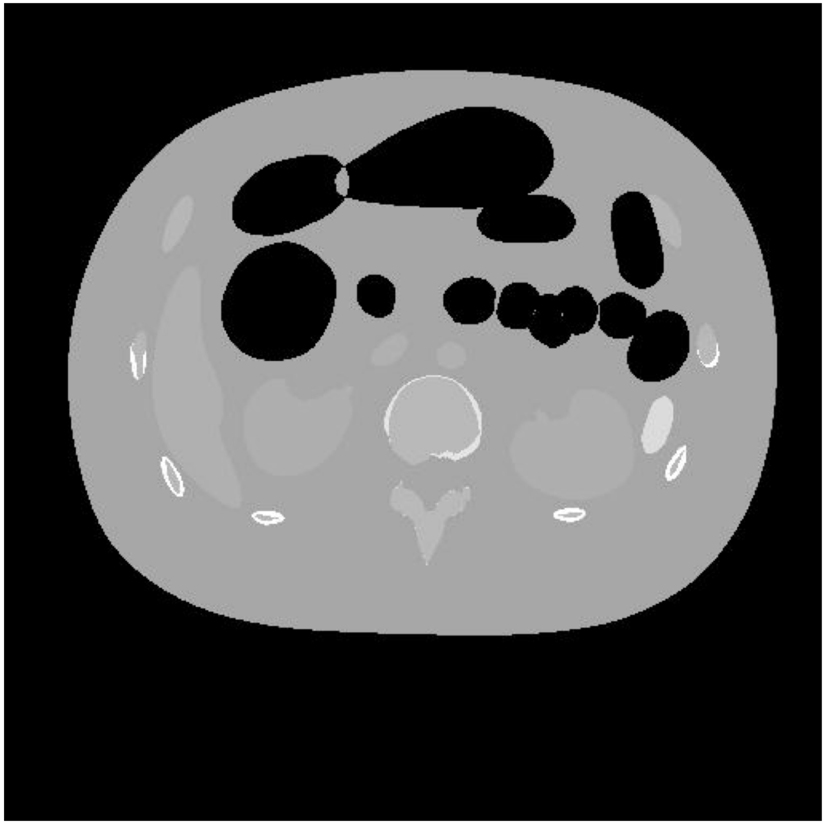
Theorem 4 (Unknown Operators in Deep Networks). *Let $\mathbf{u}_\ell(\mathbf{x}_\ell) : \mathcal{D}_\ell \rightarrow \mathcal{D}_{\ell-1}$ be a continuous function with Lipschitz-bound $l_{\mathbf{u}_\ell}$ on compact set $\mathcal{D}_\ell \subset \mathbb{R}^{N_\ell}$ with integer $\ell > 0$. Further let $\mathbf{f}_\ell(\mathbf{x}_\ell) : \mathcal{D}_\ell \rightarrow \mathcal{D}$ be a function composed of ℓ layers / function blocks defined as recursion $\mathbf{f}_\ell(\mathbf{x}_\ell) = \mathbf{f}_{\ell-1}(\mathbf{u}_\ell(\mathbf{x}_\ell))$ with $\mathbf{f}_{\ell=0}(\mathbf{x}) = \mathbf{x}$ on compact set $\mathcal{D} \subset \mathbb{R}^{N_D+1}$ bound by Lipschitz constant $l_{\mathbf{f}_\ell}$ with $l_{\mathbf{f}_{\ell=0}} = 1$. Recursive function $\hat{\mathbf{f}}_\ell(\mathbf{x}_\ell) = \hat{\mathbf{f}}_{\ell-1}(\hat{\mathbf{u}}_\ell(\mathbf{x}_\ell))$ with $\hat{\mathbf{f}}_{\ell=0}(\mathbf{x}) = \mathbf{x}$ is then an approximation of $\mathbf{f}_\ell(\mathbf{x}_\ell)$. Then, $\mathbf{e}_{f,\ell} = \mathbf{f}_\ell(\mathbf{x}_\ell) - \hat{\mathbf{f}}_\ell(\mathbf{x}_\ell)$ is generally bounded for all $\mathbf{x}_\ell \in \mathcal{D}_\ell$ and for all $\ell > 0$ in each component k by*

$$|e_{f,\ell,k}| \leq \sum_{\ell_i=1}^{\ell} ||\mathbf{e}_{u,\ell_i}||_p \cdot l_{\mathbf{f}_{\ell_i-1}} \tag{13}$$

where $\mathbf{e}_{u,\ell} = [e_{u,\ell,0}, \ldots e_{u,\ell,N_I}]^\top$ is the vector of errors introduced by $\hat{\mathbf{u}}_\ell(\mathbf{x}_\ell)$.

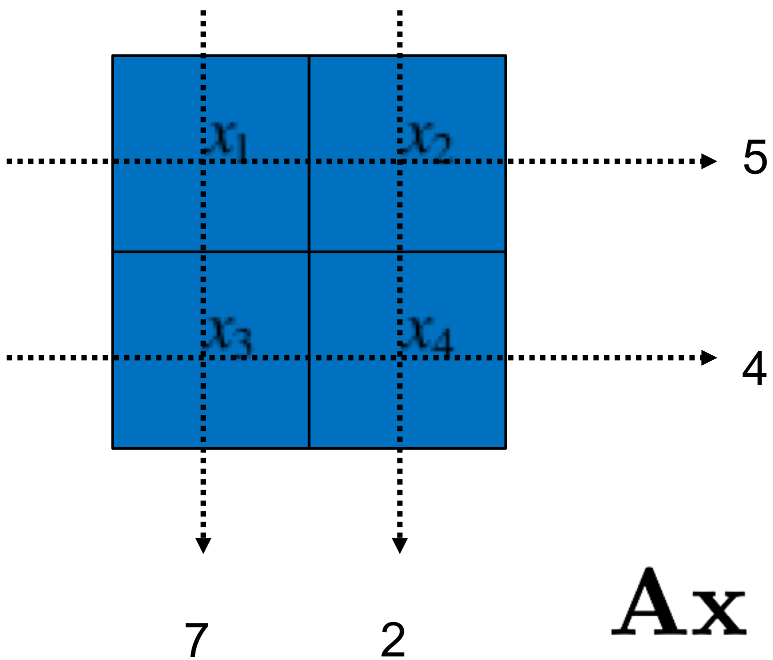
CT Reconstruction





CT Reconstruction

- CT solves a system of equations



$$\mathbf{A}\mathbf{x} = \mathbf{p}$$

$$x_1 + x_2 = 5$$
$$x_3 + x_4 = 4$$
$$x_1 + x_3 = 7$$
$$x_2 + x_4 = 2$$

$$x_1 = 3$$
$$x_2 = 2$$
$$x_3 = 4$$
$$x_4 = 0$$

CT Reconstruction

- Typically

  512 x 512 x 512 = 134 217 728

  unknown variables
- Operator A is large (about 65.000 TB in float precision)
- Efficient solution required
- Solution known since 1917
- First prototype in 1971 by Hounsfield

Computed Tomography

- Efficient solution via filtered back-projection:

$$f(x,y) = \int_0^{\pi} p(s,\theta) * \frac{1}{-2\pi^2 s^2} \mathrm{d}\theta \quad \text{mit } s = x\cos\theta + y\sin\theta$$

- Three steps:
  - Convolution along *s*
  - Back-projection along θ
  - Suppress negative values

Computed Tomography

- Efficient solution via filtered back-projection:

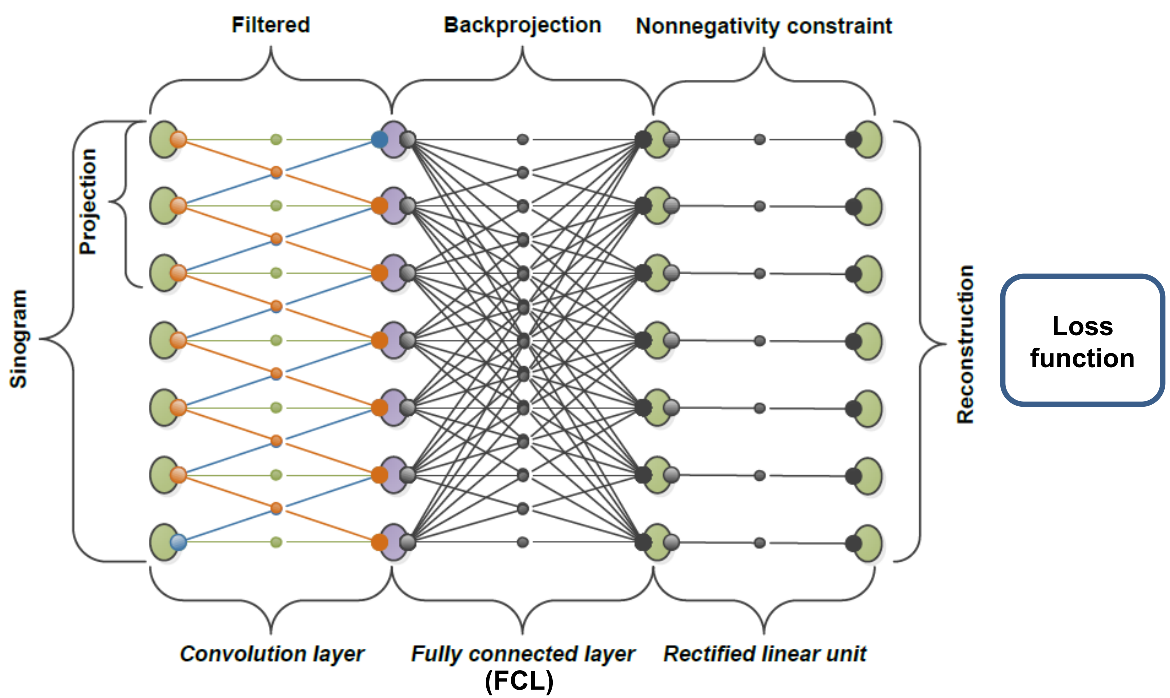
$$f(x,y) = \int_0^{\pi} p(s,\theta) * \frac{1}{-2\pi^2 s^2} \mathrm{d}\theta \quad \text{mit } s = x\cos\theta + y\sin\theta$$

- Can also be derived in matrix notation:

$$\begin{aligned} \mathbf{A}\mathbf{x} &= \mathbf{p} \\ \mathbf{x} = \mathbf{A}^{-1}\mathbf{p} &= \mathbf{A}^\top \underbrace{(\mathbf{A}\mathbf{A}^\top)^{-1}}_{\text{Filter}} \mathbf{p} \end{aligned}$$

Computed Tomography using Neural Networks

- All three steps can be modeled as neural network:



- Interesting: All weights are known from FBP

Discretization

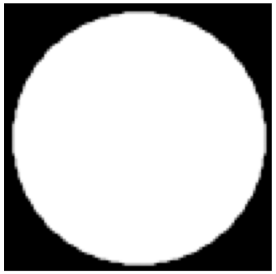
- Derivation of Radon Inverse relies on continuous form:

$$f(x,y) = \int_0^{\pi} p(s,\theta) * \frac{1}{-2\pi^2 s^2} \mathrm{d}\theta \qquad \text{mit } s = x\cos\theta + y\sin\theta$$

- Detector pixels need to be infinitely small!

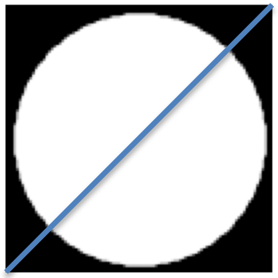
Discretization (2)

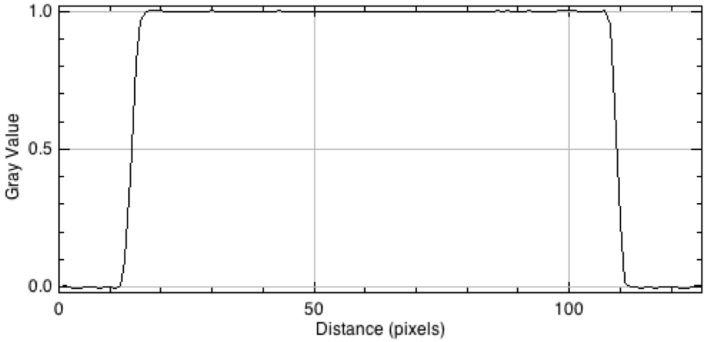
- Implementation by the book:



Discretization (2)

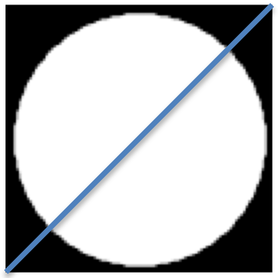
- Implementation by the book:

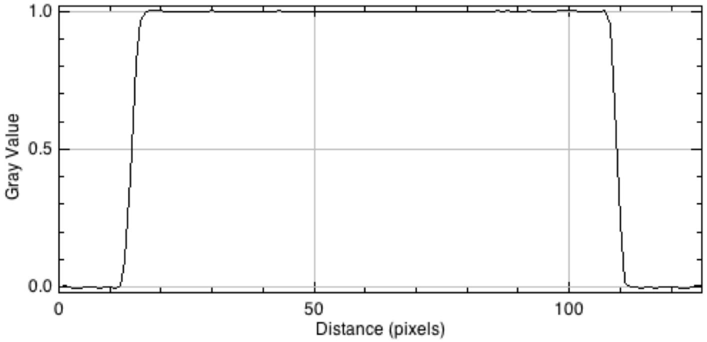




Discretization (2)

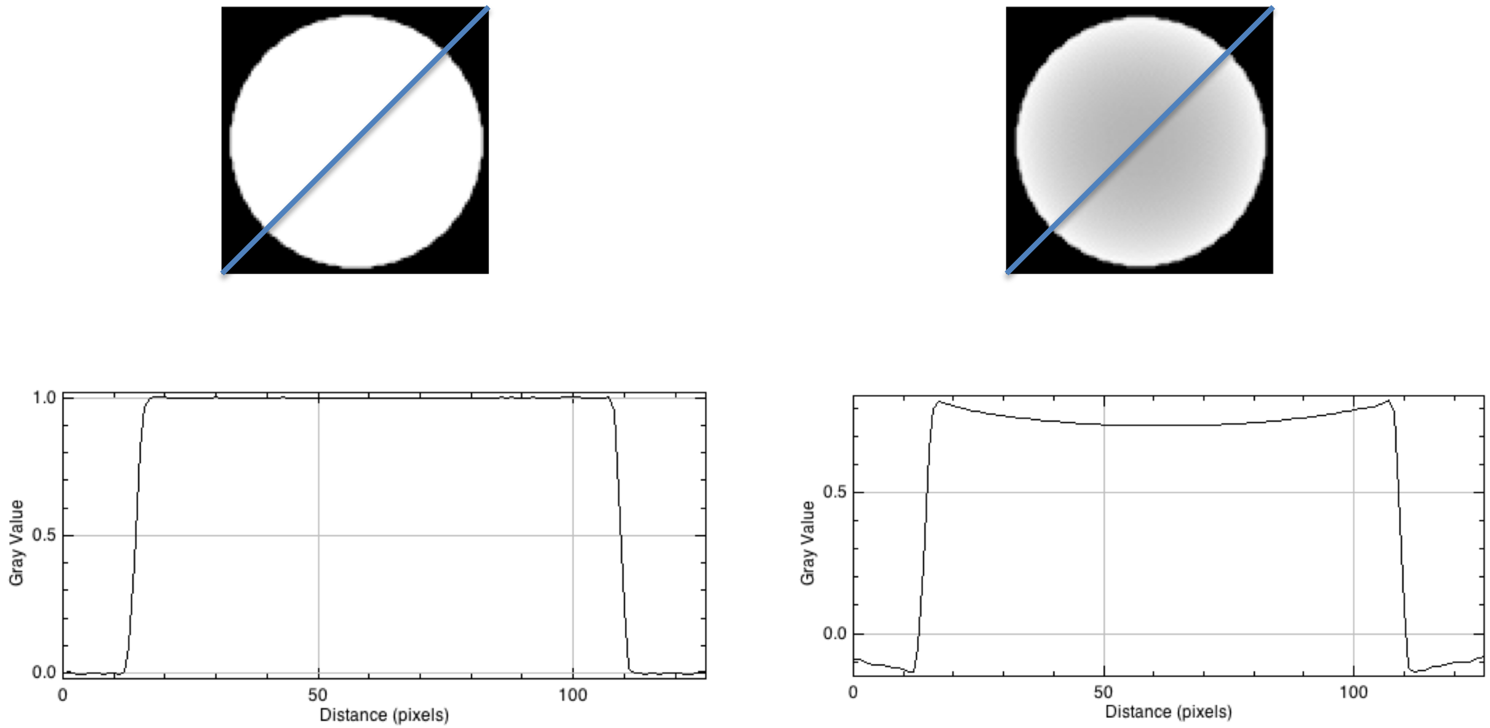
- Implementation by the book:





Discretization (2)

- Implementation by the book:



Discretization (3)

- Idea: Neural Networks are discrete end-to-end
  => Net work must learn correct solution

- Correct discretization is an intrinsic property of the net

$$\begin{aligned} \mathbf{x} &= \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top)^{-\mathbf{1}}\mathbf{p} \\ \mathbf{x} &= \mathbf{A}^\top\mathbf{F}^{\mathbf{H}}\mathbf{K}\mathbf{F}\mathbf{p} \end{aligned}$$

Discretization (3)

- Idea: Neural Networks are discrete end-to-end
  => Net work must learn correct solution

- Correct discretization is an intrinsic property of the net

$$\mathbf{x} = \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top)^{-1}\mathbf{p}$$

$$\mathbf{x} = \underbrace{\mathbf{A}^\top\mathbf{F}^\mathbf{H}\mathbf{K}\mathbf{F}}_{\text{Net}}\mathbf{p}$$

Discretization (3)

- Idea: Neural Networks are discrete end-to-end
  => Net work must learn correct solution

- Correct discretization is an intrinsic property of the net

$$\begin{aligned}\mathbf{x} &= \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top)^{-\mathbf{1}}\mathbf{p}\\ \mathbf{x} &= \mathbf{A}^\top\mathbf{F}^\mathbf{H}\mathbf{K}\mathbf{F}\mathbf{p}\end{aligned}$$

- Associated optimization problem:

$$f(\mathbf{K}) = \frac{1}{2}||\mathbf{A}^\top\mathbf{F}^\mathbf{H}\mathbf{K}\mathbf{F}\mathbf{p} - \mathbf{x}||_\mathbf{2}^\mathbf{2}$$

Discretization (3)

- Objective function:

$$f(\mathbf{K}) = \frac{1}{2}||\mathbf{A}^\top \mathbf{F}^\mathbf{H}\mathbf{K}\mathbf{F}\mathbf{p} - \mathbf{x}||_\mathbf{2}^\mathbf{2}$$

Discretization (3)

- Objective function:

$$f(\mathbf{K}) = \frac{1}{2}||\mathbf{A}^\top \mathbf{F}^\mathbf{H} \mathbf{K} \mathbf{F} \mathbf{p} - \mathbf{x}||_\mathbf{2}^\mathbf{2}$$

- Gradient:

$$\frac{\partial f(\mathbf{K})}{\partial \mathbf{K}} = \mathbf{F}\mathbf{A}(\mathbf{A}^\top \mathbf{F}^\mathbf{H} \mathbf{K} \mathbf{F} \mathbf{p} - \mathbf{x})(\mathbf{F}\mathbf{p})^\top$$

Discretization (3)

- Objective function:

$$f(\mathbf{K}) = \frac{1}{2}||\mathbf{A}^\top \mathbf{F}^\mathbf{H}\mathbf{K}\mathbf{F}\mathbf{p} - \mathbf{x}||_\mathbf{2}^\mathbf{2}$$

- Gradient:

Recon

$$\frac{\partial f(\mathbf{K})}{\partial \mathbf{K}} = \mathbf{F}\mathbf{A}(\underbrace{\mathbf{A}^\top \mathbf{F}^\mathbf{H}\mathbf{K}\mathbf{F}\mathbf{p}}_{\text{Recon}} - \mathbf{x})(\mathbf{F}\mathbf{p})^\top$$

Discretization (3)

- Objective function:

$$f(\mathbf{K}) = \frac{1}{2}||\mathbf{A}^\top \mathbf{F}^\mathbf{H}\mathbf{K}\mathbf{F}\mathbf{p} - \mathbf{x}||_\mathbf{2}^\mathbf{2}$$

- Gradient:

Error

$$\frac{\partial f(\mathbf{K})}{\partial \mathbf{K}} = \mathbf{F}\mathbf{A}(\mathbf{A}^\top \mathbf{F}^\mathbf{H}\mathbf{K}\mathbf{F}\mathbf{p} - \mathbf{x})(\mathbf{F}\mathbf{p})^\top$$

Discretization (3)

- Objective function:

$$f(\mathbf{K}) = \frac{1}{2}||\mathbf{A}^\top \mathbf{F}^\mathbf{H} \mathbf{K} \mathbf{F} \mathbf{p} - \mathbf{x}||_\mathbf{2}^\mathbf{2}$$

- Gradient:

Error

$$\frac{\partial f(\mathbf{K})}{\partial \mathbf{K}} = \mathbf{F}\mathbf{A}(\mathbf{A}^\top \mathbf{F}^\mathbf{H} \mathbf{K} \mathbf{F} \mathbf{p} - \mathbf{x})(\mathbf{F}\mathbf{p})^\top$$

Backpropagation

Discretization (3)

- Objective function:

$$f(\mathbf{K}) = \frac{1}{2}||\mathbf{A}^\top \mathbf{F}^\mathbf{H} \mathbf{K} \mathbf{F} \mathbf{p} - \mathbf{x}||_\mathbf{2}^\mathbf{2}$$

- Gradient:

$$\frac{\partial f(\mathbf{K})}{\partial \mathbf{K}} = \mathbf{F}\mathbf{A}(\mathbf{A}^\top \mathbf{F}^\mathbf{H} \mathbf{K} \mathbf{F} \mathbf{p} - \mathbf{x})(\mathbf{F}\mathbf{p})^\top$$

Backpropagation

European Research Council

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
FACULTY OF ENGINEERING

Discretization (3)

- Objective function:

$$f(\mathbf{K}) = \frac{1}{2}||\mathbf{A}^\top \mathbf{F}^\mathbf{H} \mathbf{K} \mathbf{F} \mathbf{p} - \mathbf{x}||_\mathbf{2}^\mathbf{2}$$

- Gradient:

$$\frac{\partial f(\mathbf{K})}{\partial \mathbf{K}} = \mathbf{FA}(\mathbf{A}^\top \mathbf{F}^\mathbf{H} \mathbf{K} \mathbf{F} \mathbf{p} - \mathbf{x})(\mathbf{F}\mathbf{p})^\top$$

Backpropagation „l-1“

Discretization (3)

- Objective function:

$$f(\mathbf{K}) = \frac{1}{2}||\mathbf{A}^\top \mathbf{F}^\mathbf{H} \mathbf{K} \mathbf{F} \mathbf{p} - \mathbf{x}||_2^2$$

- Gradient:

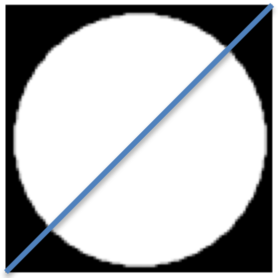
$$\frac{\partial f(\mathbf{K})}{\partial \mathbf{K}} = \mathbf{F} \mathbf{A} (\mathbf{A}^\top \mathbf{F}^\mathbf{H} \mathbf{K} \mathbf{F} \mathbf{p} - \mathbf{x}) (\mathbf{F} \mathbf{p})^\top$$

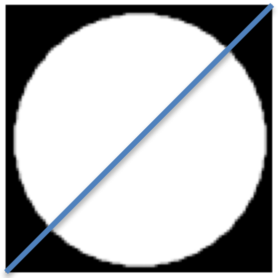
Backpropagation „l-1“

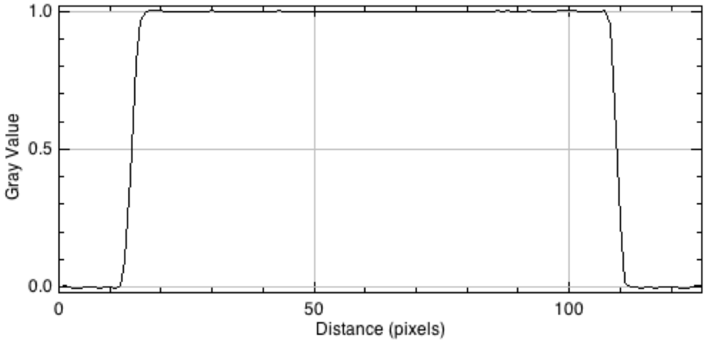
$$\Delta w_{ij}^{(l)} = \eta \, \delta_j^{(l)} \, y_i^{(l-1)}$$

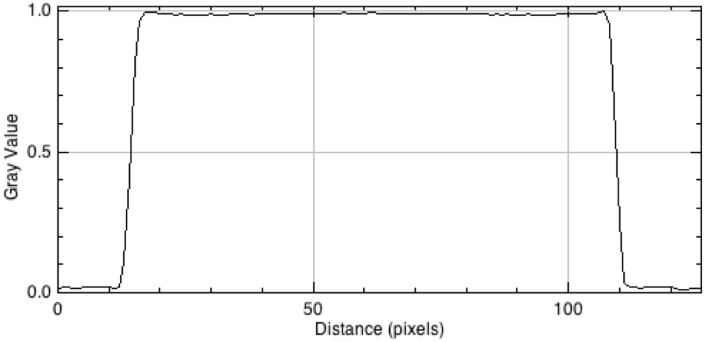
Discretization (4)

- Filter after „Learning“:





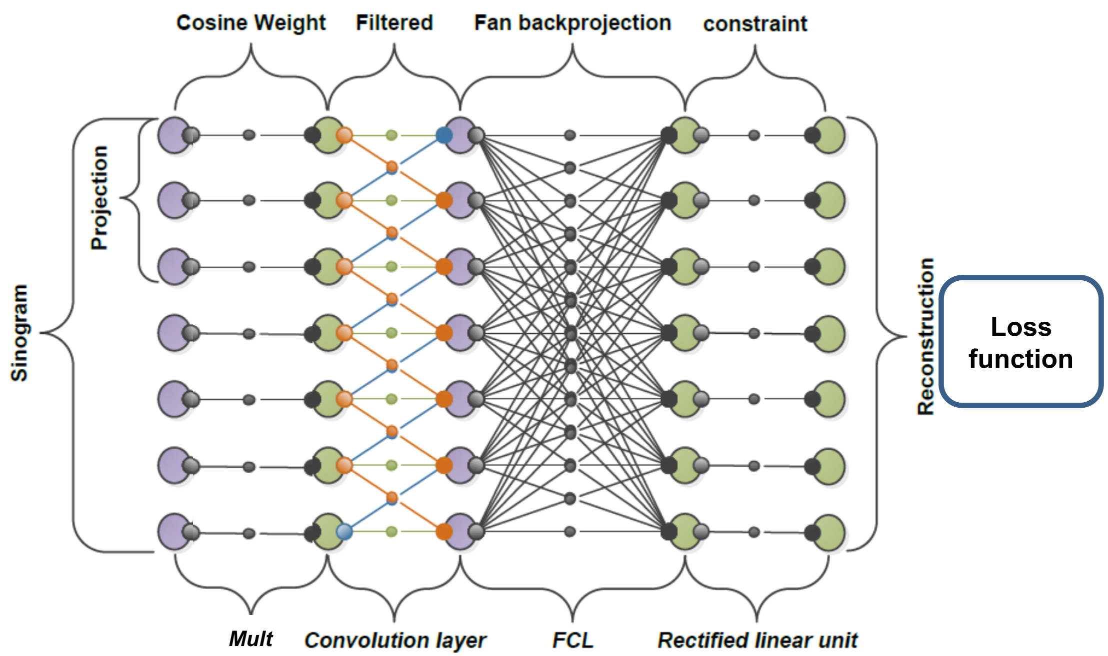




Computed Tomography using Neural Networks

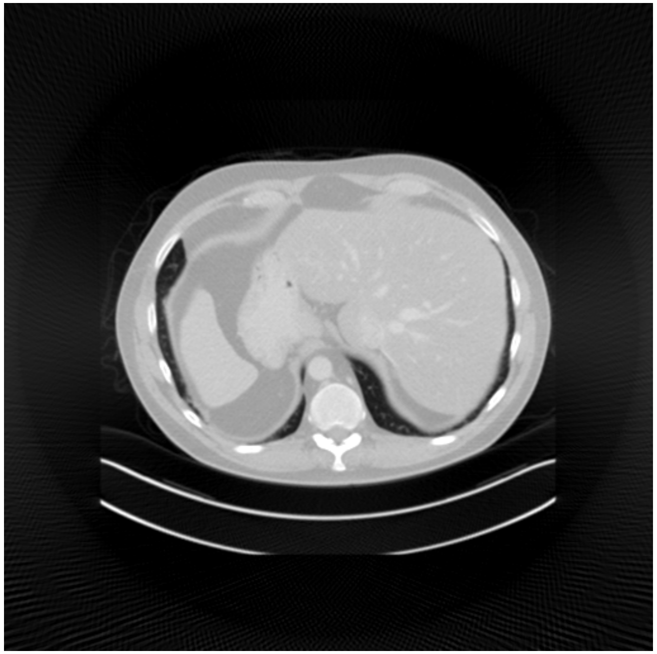
- Fan-beam reconstruction formula:

$$x = \mathbf{A}^\top \mathbf{C}\mathbf{W}\mathbf{p}$$



Computed Tomography using Neural Networks

- Application to incomplete scans [2]

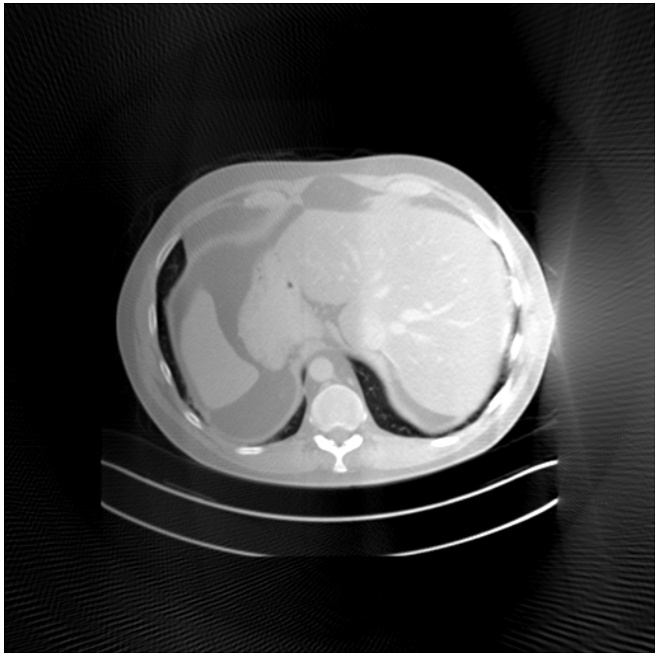


Reconstruction with 360 deg

[6] Tobias Würfl, Florin Ghesu, Vincent Christlein, Andreas Maier. Deep Learning Computed Tomography. MICCAI 2016.

Computed Tomography using Neural Networks

- **Application to incomplete scans [2]**

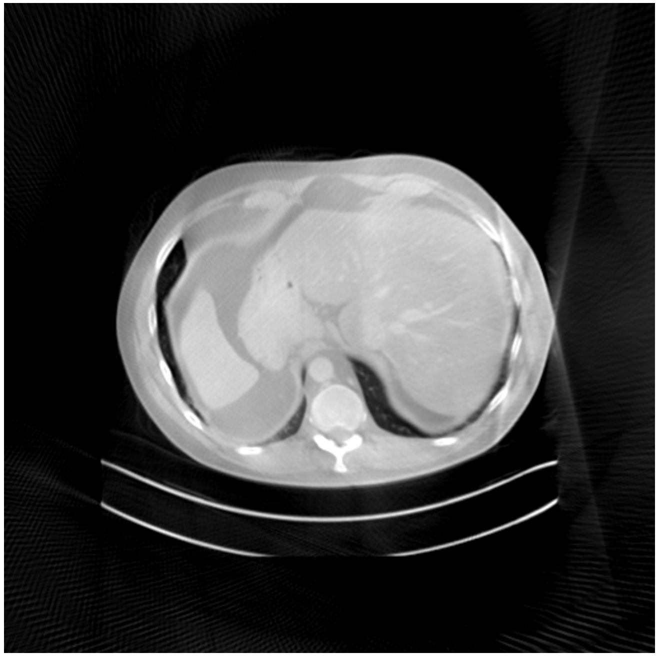


Reconstruction with 180 deg (FBP)

[6] Tobias Würfl, Florin Ghesu, Vincent Christlein, Andreas Maier. Deep Learning Computed Tomography. MICCAI 2016.

Computed Tomography using Neural Networks

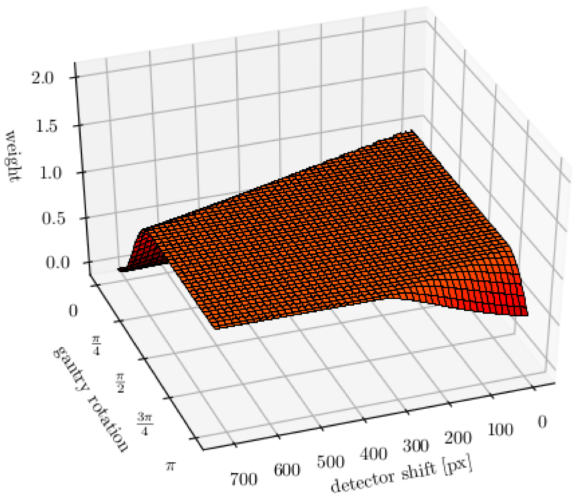
- Application to incomplete scans [2]



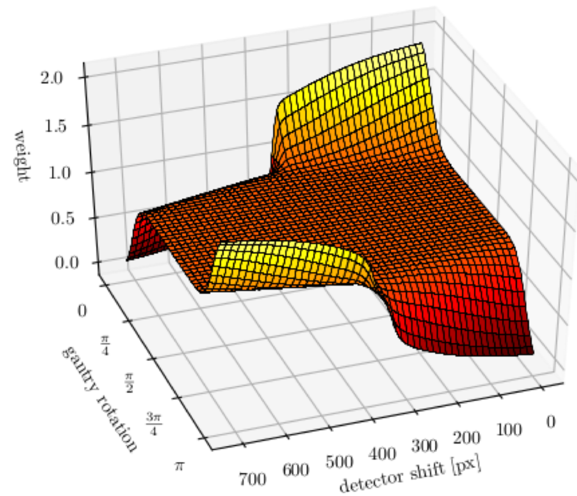
Reconstruction with 180 deg (NN)

[6] Tobias Würfl, Florin Ghesu, Vincent Christlein, Andreas Maier. Deep Learning Computed Tomography. MICCAI 2016.

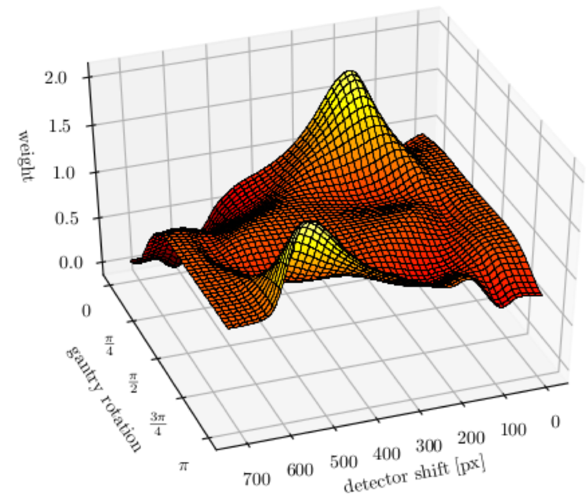
Learned Weights



Parker et al.
1982
“works well“



Schäfer et al.
2017
„works better“



N. N. et al.
2016
data optimal

[6] Tobias Würfl, Florin Ghesu, Vincent Christlein, Andreas Maier. Deep Learning Computed Tomography. MICCAI 2016.

Further Extensions

- **Add non-linear de-streaking and de-noising step:**

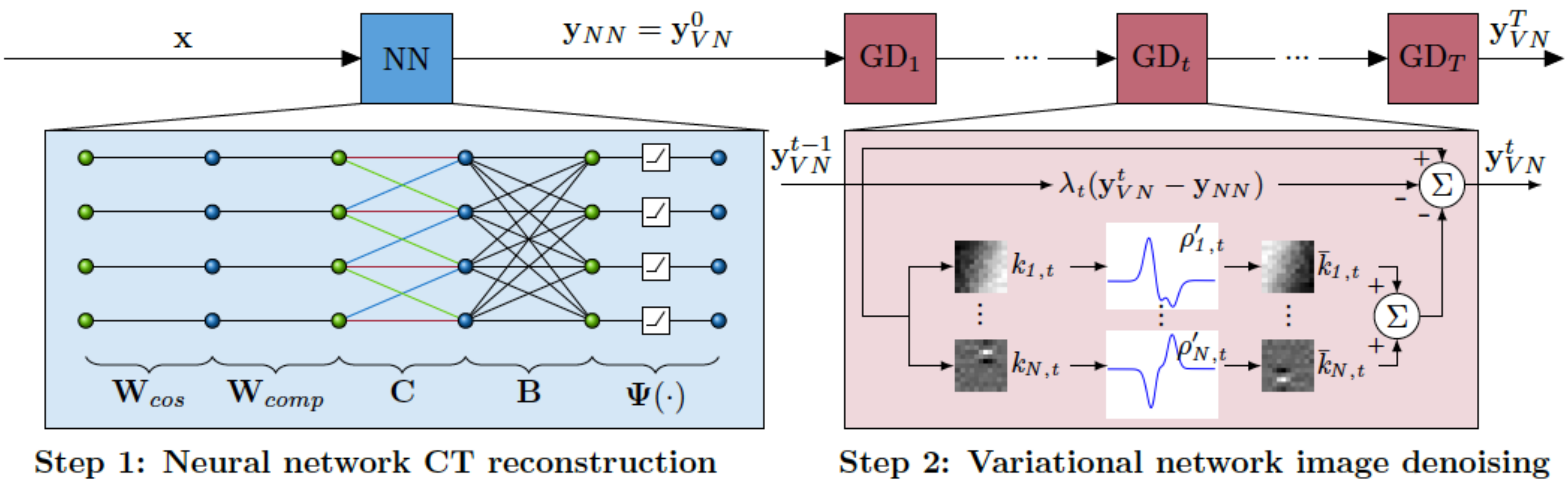
$$E(\boldsymbol{y}) = \frac{\lambda}{2}\|\boldsymbol{y}_{VN} - \boldsymbol{y}_{NN}\|_2^2 + \sum_{i=1}^{N_k} \rho_i(\mathbf{K}_i \boldsymbol{y}_{VN})$$

$$\boldsymbol{y}_{VN}^{t} = \boldsymbol{y}_{VN}^{t-1} - \sum_{i=1}^{N_k} \mathbf{K}_{i,t}^{T} \rho'_{i,t}(\mathbf{K}_{i,t} \boldsymbol{y}_{VN}^{t-1}) - \lambda_t(\boldsymbol{y}_{VN}^{t-1} - \boldsymbol{y}_{NN})$$

[7] Hammernik, Kerstin, et al. "A deep learning architecture for limited-angle computed tomography reconstruction." *Bildverarbeitung für die Medizin 2017*. Springer Vieweg, Berlin, Heidelberg, 2017. 92-97.

Further Extensions

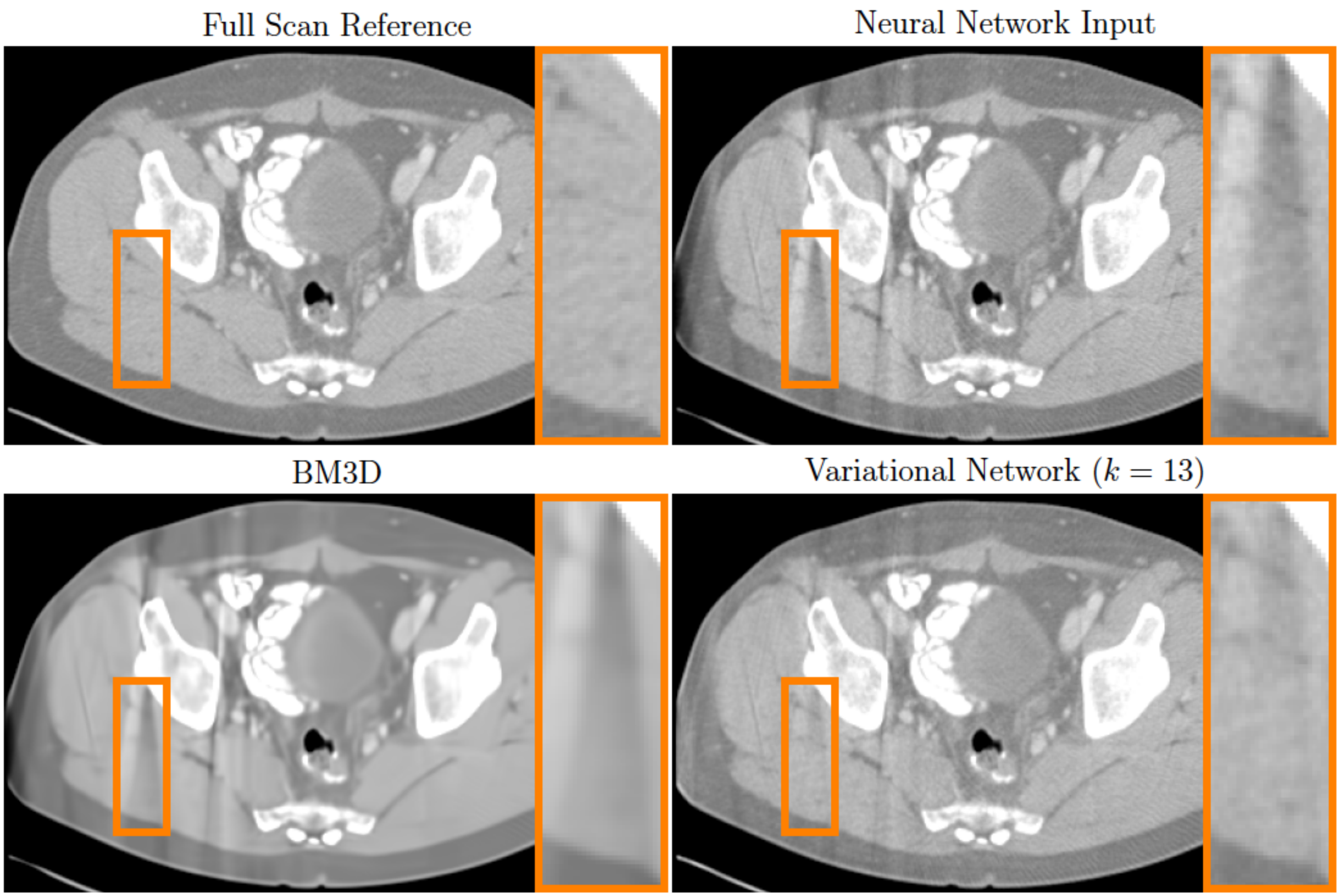
- **Add non-linear de-streaking and de-noising step:**



[7] Hammernik, Kerstin, et al. "A deep learning architecture for limited-angle computed tomography reconstruction." *Bildverarbeitung für die Medizin 2017*. Springer Vieweg, Berlin, Heidelberg, 2017. 92-97.

Lehrstuhl für Mustererkennung / FAU Friedrich-Alexander-Universität Erlangen-Nürnberg, Faculty of Engineering; European Research Council

Further Extensions



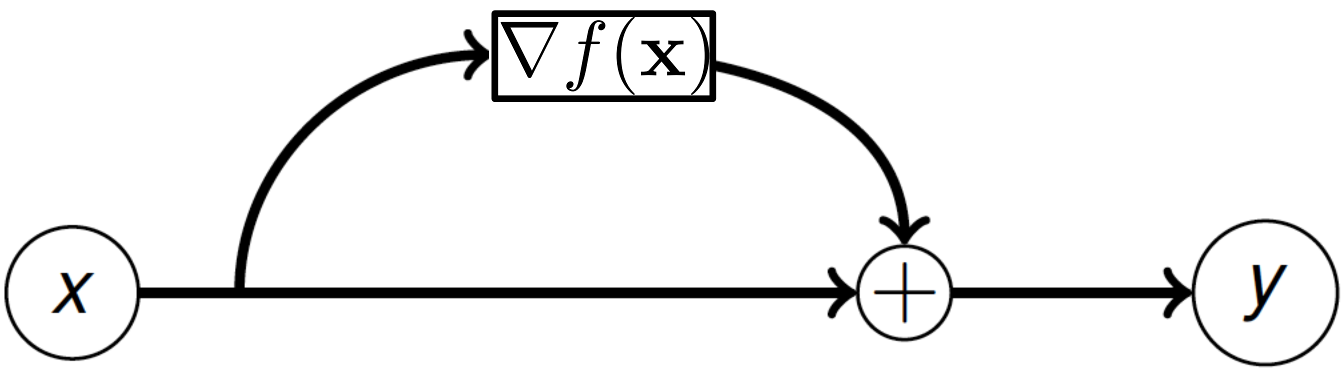
ResNets Revisited

General Function Optimization: Find maxima of

$$f(\mathbf{x})$$

Idea: follow gradient direction

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \nabla f(\mathbf{x}_n)$$



European Research Council

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
FACULTY OF ENGINEERING

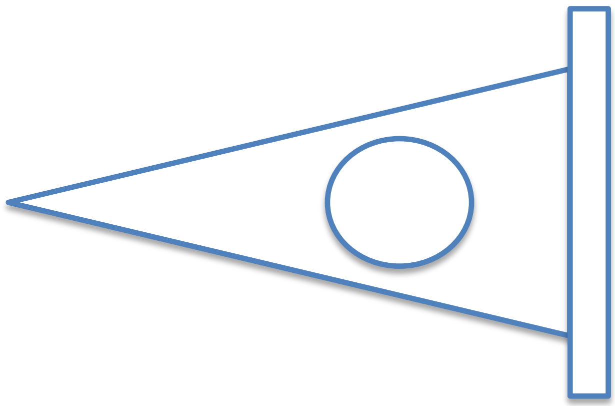
Can we „derive“ networks?

$$\mathbf{A}_{CB}\mathbf{x} = \mathbf{p}_{CB}$$

[9] Christopher Syben, Bernhard Stimpel, Jonathan Lommen, Tobias Würfl, Arnd Dörfler, Andreas Maier. Deriving Neural Network Architectures using Precision Learning: Parallel-to-fan beam Conversion. GCPR 2018. https://arxiv.org/abs/1807.03057

Can we „derive“ networks?

$$\mathbf{A}_{CB}\mathbf{x} = \mathbf{p}_{CB}$$



Cone-beam acquisition

[9] Christopher Syben, Bernhard Stimpel, Jonathan Lommen, Tobias Würfl, Arnd Dörfler, Andreas Maier. Deriving Neural Network Architectures using Precision Learning: Parallel-to-fan beam Conversion. GCPR 2018. https://arxiv.org/abs/1807.03057

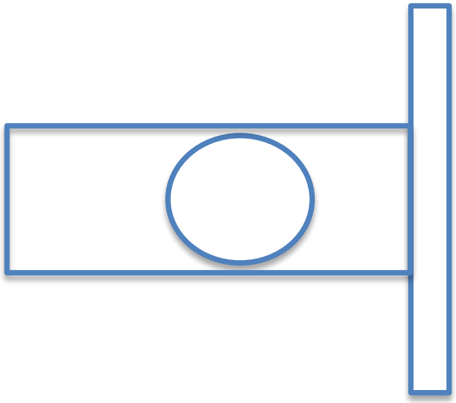
Can we „derive“ networks?

$$\mathbf{A}_{CB}\mathbf{x} = \mathbf{p}_{CB}$$
$$\mathbf{A}_{PB}\mathbf{x} = \mathbf{p}_{PB}$$

[9] Christopher Syben, Bernhard Stimpel, Jonathan Lommen, Tobias Würfl, Arnd Dörfler, Andreas Maier. Deriving Neural Network Architectures using Precision Learning: Parallel-to-fan beam Conversion. GCPR 2018. https://arxiv.org/abs/1807.03057

Can we „derive“ networks?

$$\mathbf{A}_{CB}\mathbf{x} = \mathbf{p}_{CB}$$
$$\mathbf{A}_{PB}\mathbf{x} = \mathbf{p}_{PB}$$

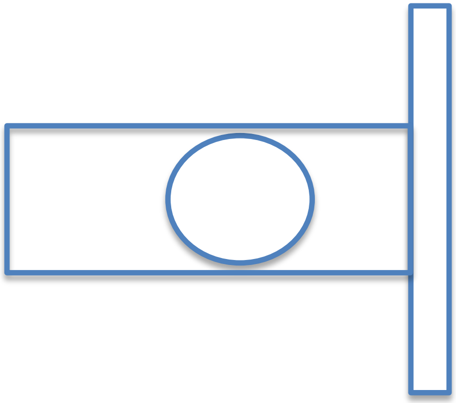


Parallel projection

[9] Christopher Syben, Bernhard Stimpel, Jonathan Lommen, Tobias Würfl, Arnd Dörfler, Andreas Maier. Deriving Neural Network Architectures using Precision Learning: Parallel-to-fan beam Conversion. GCPR 2018. https://arxiv.org/abs/1807.03057

Can we „derive“ networks?

$$\mathbf{A}_{CB}\mathbf{x} = \mathbf{p}_{CB}$$
$$\mathbf{A}_{PB}\mathbf{x} = \mathbf{p}_{PB}$$



Parallel projection
Can't be measured!

[9] Christopher Syben, Bernhard Stimpel, Jonathan Lommen, Tobias Würfl, Arnd Dörfler, Andreas Maier. Deriving Neural Network Architectures using Precision Learning: Parallel-to-fan beam Conversion. GCPR 2018. https://arxiv.org/abs/1807.03057

Can we „derive“ networks?

$$\begin{aligned} \mathbf{A}_{CB}\mathbf{x} &= \mathbf{p}_{CB} \\ \mathbf{A}_{PB}\mathbf{x} &= \mathbf{p}_{PB} \\ x &= \mathbf{A}_{CB}^{\top}(\mathbf{A}_{CB}\mathbf{A}_{CB}^{\top})^{-1}\mathbf{p}_{CB} \end{aligned}$$

[9] Christopher Syben, Bernhard Stimpel, Jonathan Lommen, Tobias Würfl, Arnd Dörfler, Andreas Maier. Deriving Neural Network Architectures using Precision Learning: Parallel-to-fan beam Conversion. GCPR 2018. https://arxiv.org/abs/1807.03057

Can we „derive“ networks?

$$
\begin{aligned}
\mathbf{A}_{CB}\mathbf{x} &= \mathbf{p}_{CB} \\
\mathbf{A}_{PB}\mathbf{x} &= \mathbf{p}_{PB} \\
x &= \mathbf{A}_{CB}^{\top}(\mathbf{A}_{CB}\mathbf{A}_{CB}^{\top})^{-1}\mathbf{p}_{CB} \\
\mathbf{p}_{PB} &= \mathbf{A}_{PB}\mathbf{A}_{CB}^{\top}(\mathbf{A}_{CB}\mathbf{A}_{CB}^{\top})^{-1}\mathbf{p}_{CB}
\end{aligned}
$$

[9] Christopher Syben, Bernhard Stimpel, Jonathan Lommen, Tobias Würfl, Arnd Dörfler, Andreas Maier. Deriving Neural Network Architectures using Precision Learning: Parallel-to-fan beam Conversion. GCPR 2018. https://arxiv.org/abs/1807.03057

Can we „derive“ networks?

$$
\begin{aligned}
\mathbf{A}_{CB}\mathbf{x} &= \mathbf{p}_{CB} \\
\mathbf{A}_{PB}\mathbf{x} &= \mathbf{p}_{PB} \\
x &= \mathbf{A}_{CB}^{\top}(\mathbf{A}_{CB}\mathbf{A}_{CB}^{\top})^{-1}\mathbf{p}_{CB} \\
\mathbf{p}_{PB} &= \mathbf{A}_{PB}\mathbf{A}_{CB}^{\top}(\mathbf{A}_{CB}\mathbf{A}_{CB}^{\top})^{-1}\mathbf{p}_{CB} \\
\mathbf{p}_{PB} &= \mathbf{A}_{PB}\mathbf{A}_{CB}^{\top}F^{H}KF\mathbf{p}_{CB}
\end{aligned}
$$

[9] Christopher Syben, Bernhard Stimpel, Jonathan Lommen, Tobias Würfl, Arnd Dörfler, Andreas Maier. Deriving Neural Network Architectures using Precision Learning: Parallel-to-fan beam Conversion. GCPR 2018. https://arxiv.org/abs/1807.03057

Can we „derive“ networks?

$$\mathbf{A}_{CB}\mathbf{x} = \mathbf{p}_{CB}$$

$$\mathbf{A}_{PB}\mathbf{x} = \mathbf{p}_{PB}$$

$$x = \mathbf{A}_{CB}^{\top}(\mathbf{A}_{CB}\mathbf{A}_{CB}^{\top})^{-1}\mathbf{p}_{CB}$$

$$\mathbf{p}_{PB} = \mathbf{A}_{PB}\mathbf{A}_{CB}^{\top}(\mathbf{A}_{CB}\mathbf{A}_{CB}^{\top})^{-1}\mathbf{p}_{CB}$$

$$\mathbf{p}_{PB} = \mathbf{A}_{PB}\mathbf{A}_{CB}^{\top}F^{H}KF\mathbf{p}_{CB}$$

New Net Topology ?

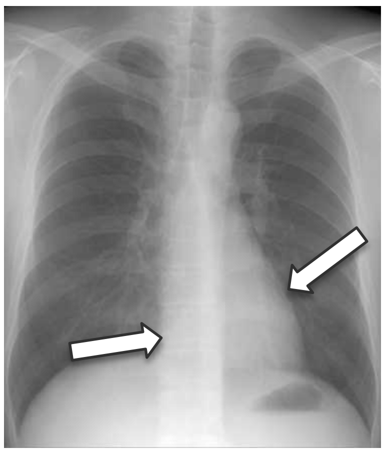
[9] Christopher Syben, Bernhard Stimpel, Jonathan Lommen, Tobias Würfl, Arnd Dörfler, Andreas Maier. Deriving Neural Network Architectures using Precision Learning: Parallel-to-fan beam Conversion. GCPR 2018. https://arxiv.org/abs/1807.03057

Known Operator Learning

- Introduction
- Current State-of-the-art in Deep Learning
- Prior Operators in Deep Networks
- Future Work

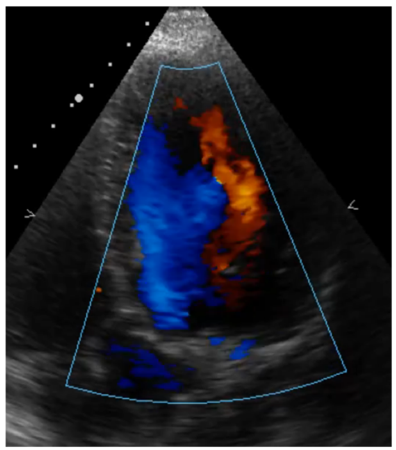
4D+ nanoSCOPE Project

Cardiology



X-ray micrograph

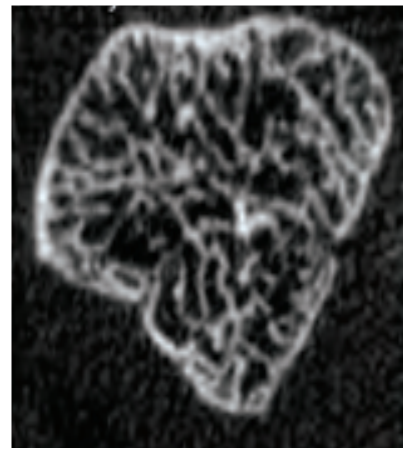
shape



Duplex-Sonography*

function

Bone biology



X-ray computed tomography

shape

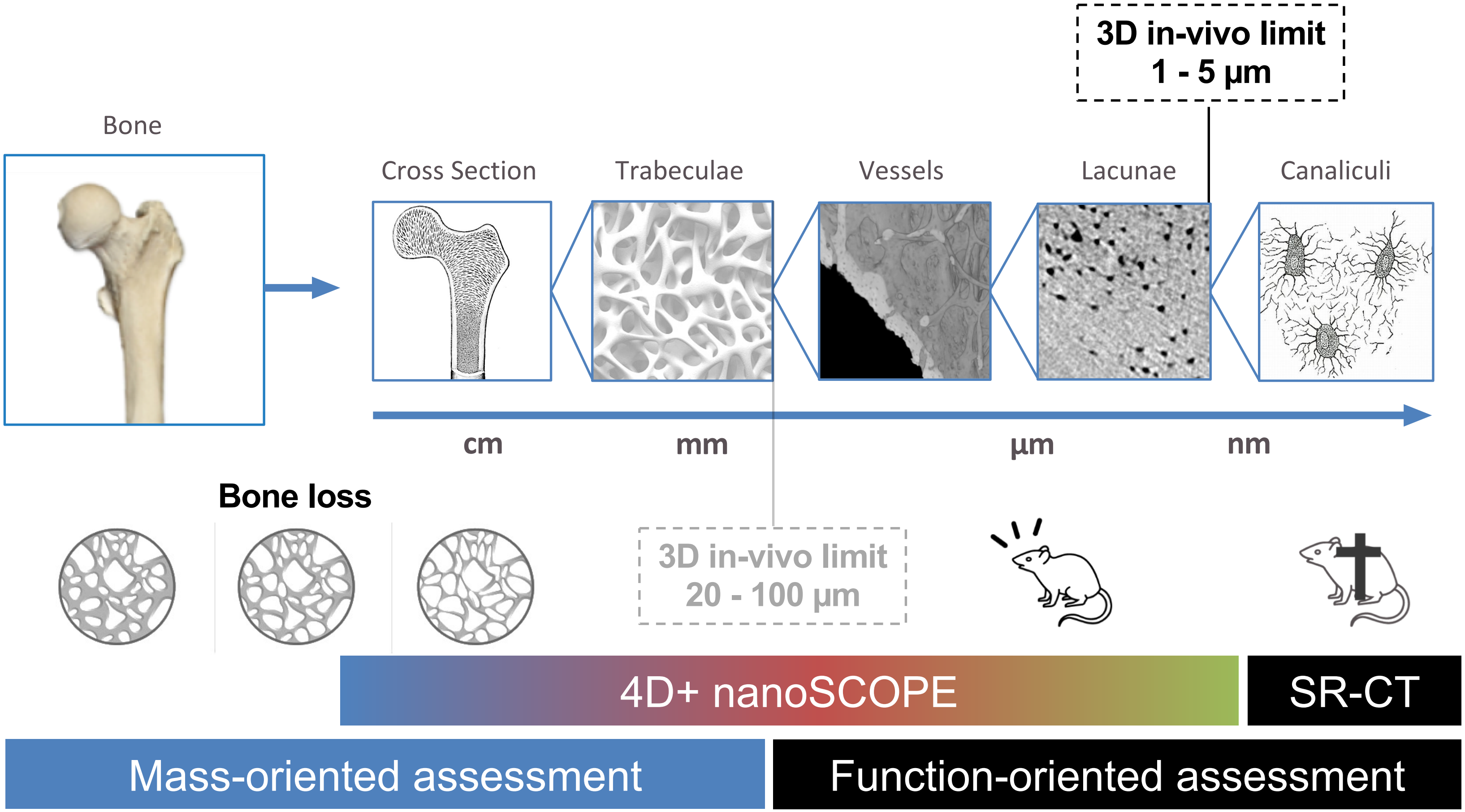
4D+ NANOSCOPE

In-vivo x-ray bone microscopy

function

**Kindly provided by S. Achenbach, FAU Erlangen*

Breaking present limits of bone analysis



Muller et al., Hierarchical microimaging of bone structure and function, Nat. Rev. Rheum. 5, 373 (2009)

Known Operator Learning

- Many traditional approaches mathematically equivalent to neural networks and vice versa
- Learned algorithms are again traditional algorithms
- Learned parameters can be interpreted!
- Virtually all state-of-the art methods can also be integrated
- Methods efficient and interpretable

Thank you for your attention!

[1] Florin Ghesu et al. Robust Multi-Scale Anatomical Landmark Detection in Incomplete 3D-CT Data. Medical Image Computing and Computer-Assisted Intervention MICCAI 2017 (MICCAI), Quebec, Canada, pp. 194-202, 2017 – **MICCAI Young Researcher Award**

[2] Florin Ghesu et al. Multi-Scale Deep Reinforcement Learning for Real-Time 3D-Landmark Detection in CT Scans. IEEE Transactions on Pattern Analysis and Machine Intelligence. ePub ahead of print. 2018

[3] Bastian Bier et al. X-ray-transform Invariant Anatomical Landmark Detection for Pelvic Trauma Surgery. MICCAI 2018 – **MICCAI Young Researcher Award**

[4] Yixing Huang et al. Some Investigations on Robustness of Deep Learning in Limited Angle Tomography. MICCAI 2018.

[5] Andreas Maier et al. Precision Learning: Towards use of known operators in neural networks. ICPR 2018.

[6] Tobias Würfl, Florin Ghesu, Vincent Christlein, Andreas Maier. Deep Learning Computed Tomography. MICCAI 2016.

[7] Hammernik, Kerstin, et al. "A deep learning architecture for limited-angle computed tomography reconstruction." *Bildverarbeitung für die Medizin 2017*. Springer Vieweg, Berlin, Heidelberg, 2017. 92-97.

[8] Aubreville, Marc, et al. "Deep Denoising for Hearing Aid Applications." 2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC). IEEE, 2018.

[9] Christopher Syben, Bernhard Stimpel, Jonathan Lommen, Tobias Würfl, Arnd Dörfler, Andreas Maier. Deriving Neural Network Architectures using Precision Learning: Parallel-to-fan beam Conversion. GCPR 2018. https://arxiv.org/abs/1807.03057

[10] Stromer, Daniel, et al. "Browsing through sealed historical manuscripts by using 3-D computed tomography with low-brilliance X-ray sources." Scientific reports 8.1 (2018): 15335.

[11] Stromer, Daniel, et al. "Virtual cleaning and unwrapping of non-invasively digitized soiled bamboo scrolls." *Scientific reports* 9.1 (2019): 2311.