

# Deep Image and Deep Dose Formation in CT



**Marc Kachelrieß**

**German Cancer Research Center (DKFZ)**

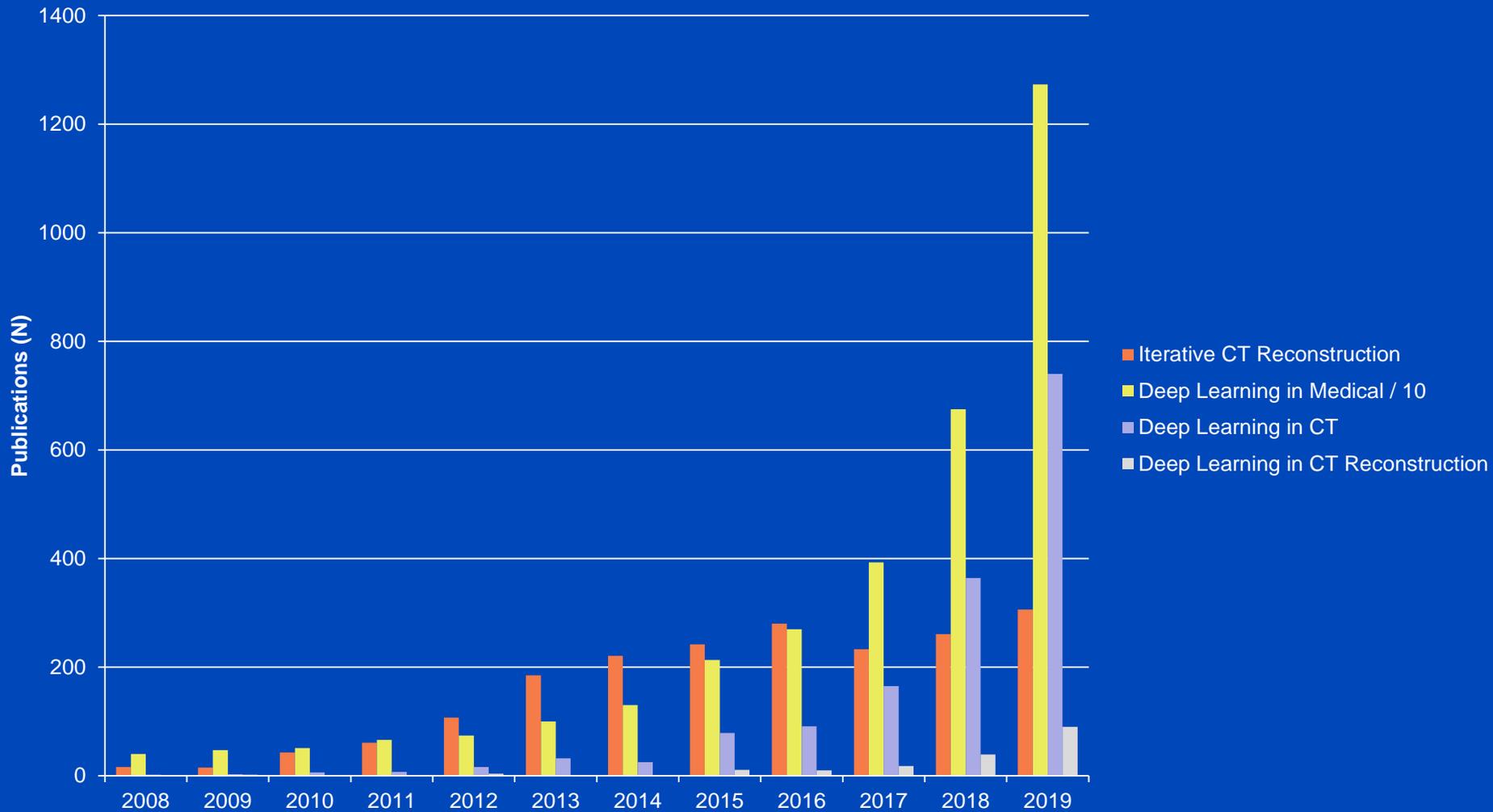
**Heidelberg, Germany**

**[www.dkfz.de/ct](http://www.dkfz.de/ct)**



**DEUTSCHES  
KREBSFORSCHUNGSZENTRUM  
IN DER HELMHOLTZ-GEMEINSCHAFT**

# Overview Publications in PubMed



2019 estimated for the whole year based on the values as of July 17, 2019.

# Fully Connected Neural Network

- Each layer fully connects to previous layer
- Difficult to train (many parameters in  $W$  and  $b$ )
- Spatial relations not necessarily preserved

Input

Hidden

Hidden

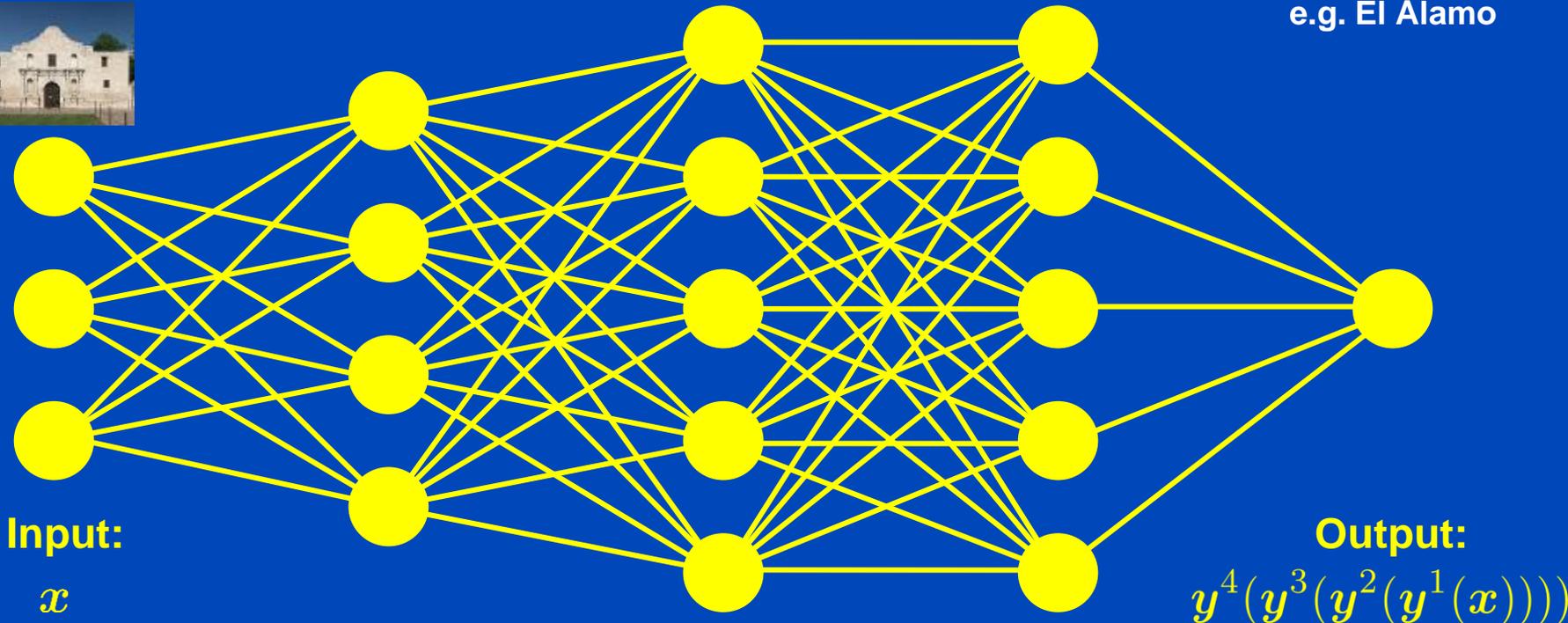
Hidden

Output

e.g. 75x100x3 pixels  
e.g.



e.g. 1 label  
e.g. El Alamo



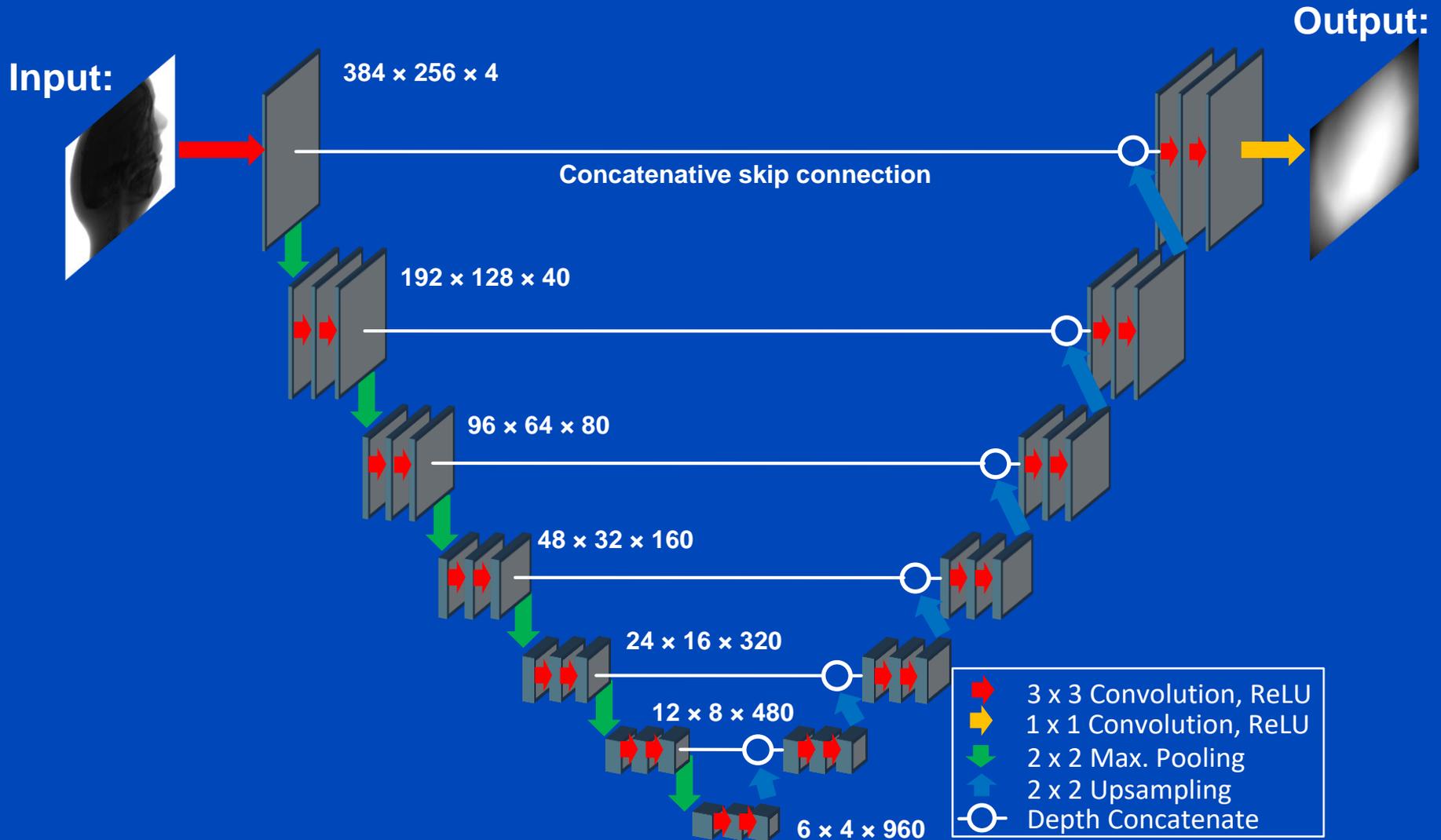
$y(x) = f(W \cdot x + b)$  with  $f(x) = (f(x_1), f(x_2), \dots)$  point-wise scalar, e.g.  $f(x) = x \vee 0 = \text{ReLU}$

# Convolutional Neural Network (CNN)

- Replace dense  $W$  in  $y(x) = f(W \cdot x + b)$  by a sparse matrix  $W$  with sparsity being of convolutional type.
- CNNs consist (mainly) of convolutional layers.
- Convolutional layers are not fully connected.
- Convolutional layers are connected by small, say  $3 \times 3$ , convolution kernels whose entries need to be found by training.
- CNNs preserve spatial relations to some extent.



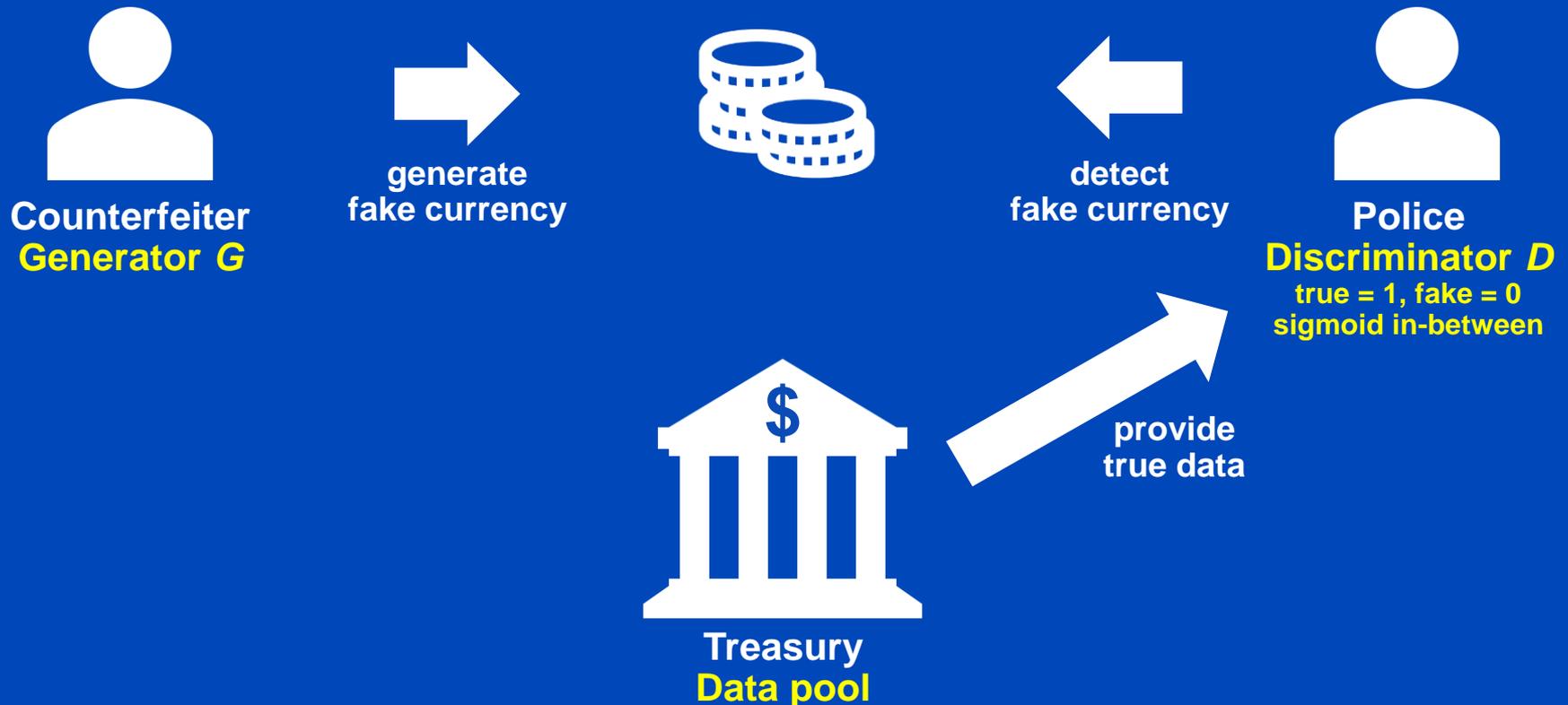
# U-Net<sup>1</sup>



<sup>1</sup>O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. Proc. MICCAI:234-241, 2015.

# Generative Adversarial Network<sup>1</sup> (GAN)

- Useful, if no direct ground truth (GT) is available, the training data are unpaired, unsupervised learning



<sup>1</sup>Goodfellow et al. 2014

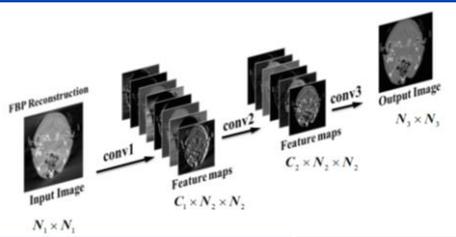
# Outline

1. Making up data
2. Noise reduction
3. Replacement of lengthy computations
4. Image reconstruction

# Part 1:

# Making up Data

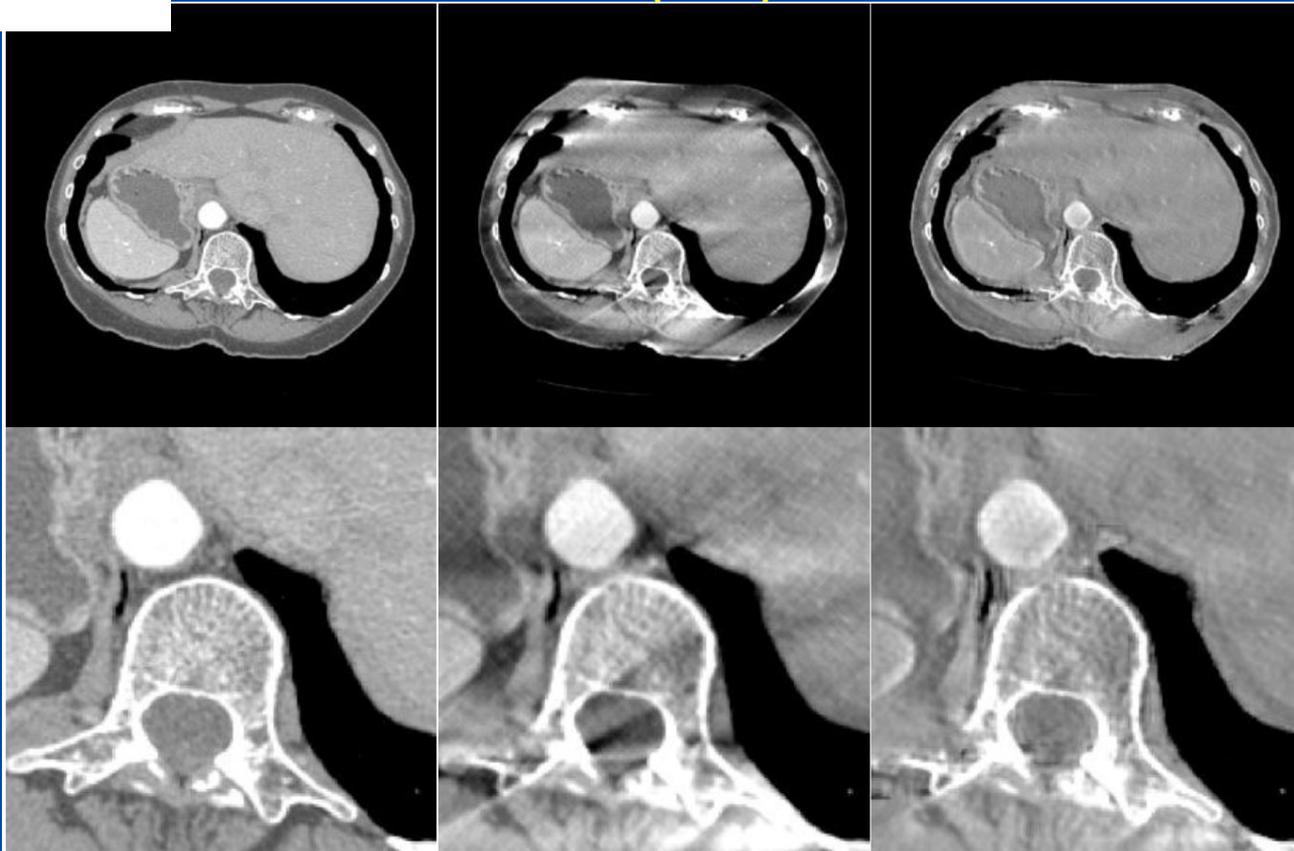
# Limited Angle Example



**GT**

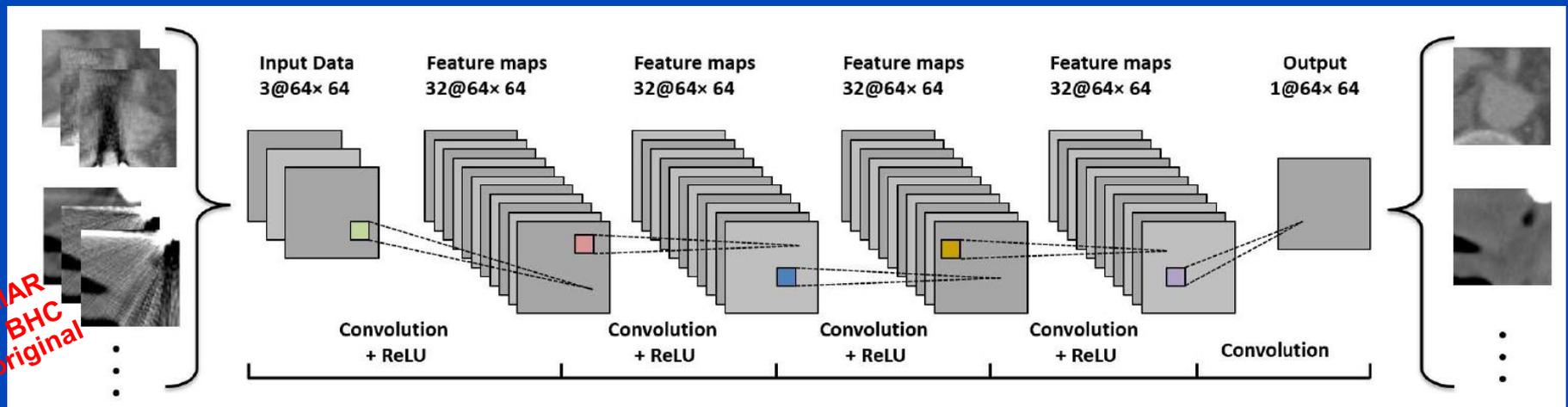
**FBP (150°)**

**CNN**

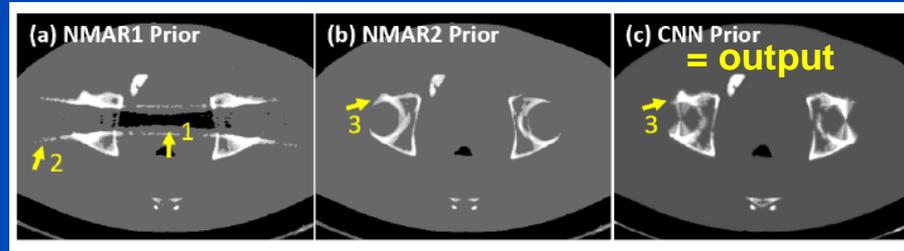
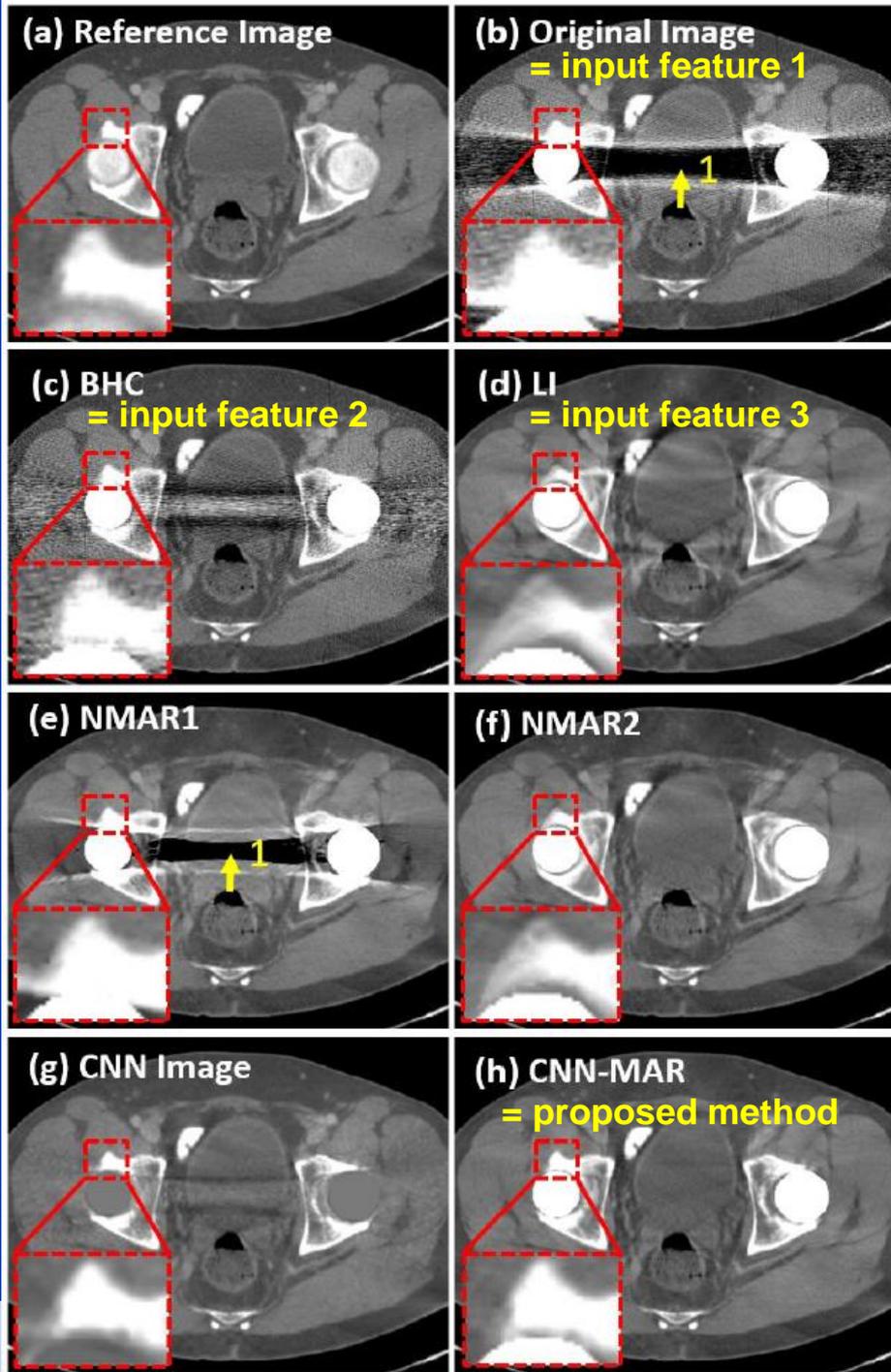


# MAR Example

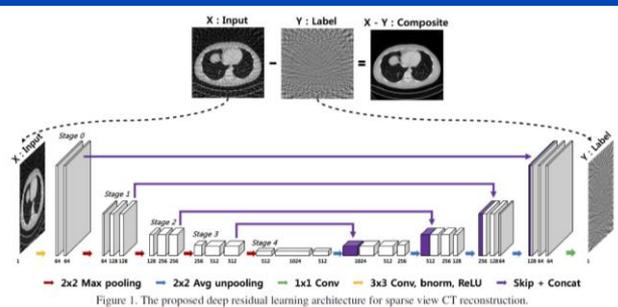
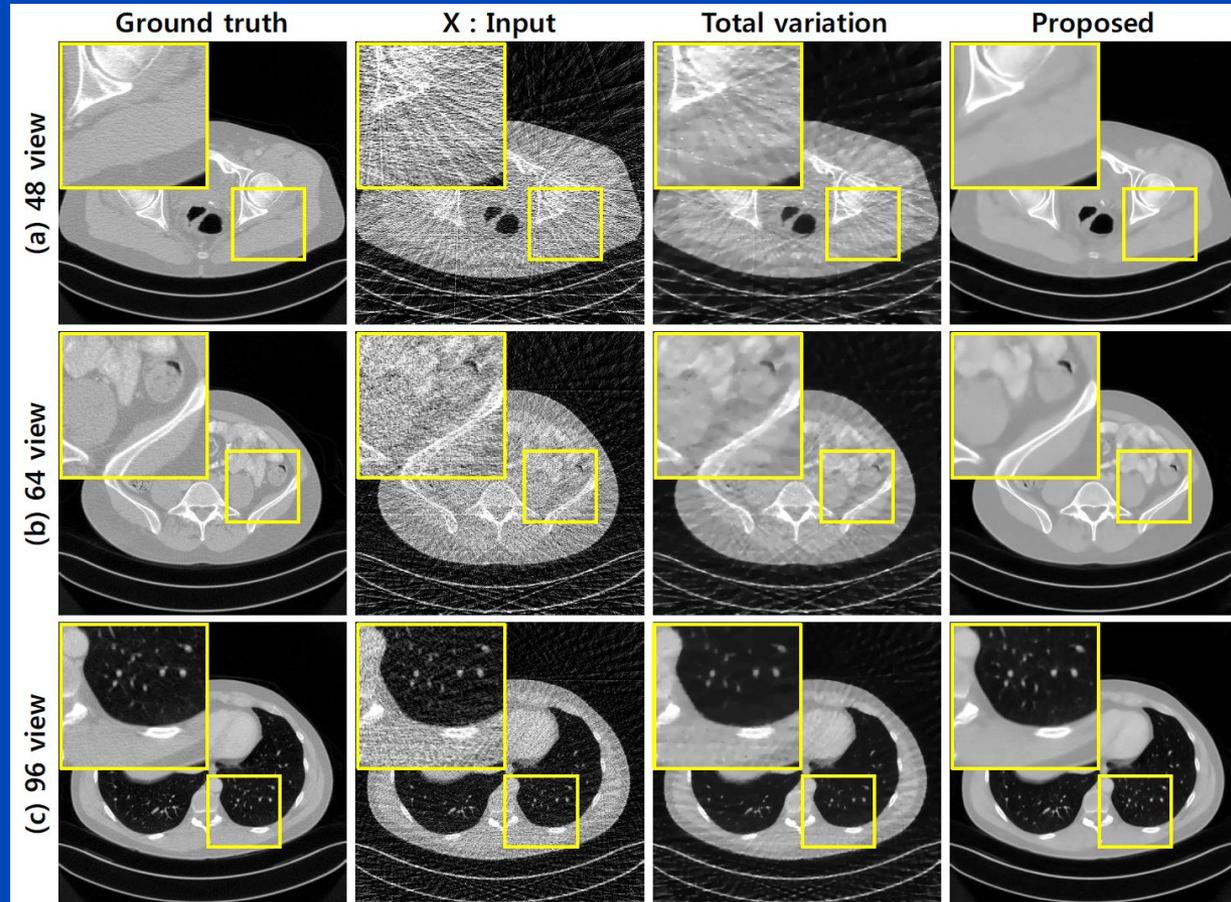
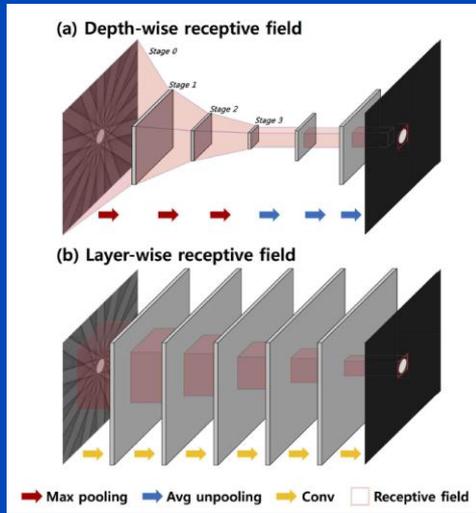
- Deep CNN-driven patch-based combination of the advantages of several MAR methods trained on simulated artifacts

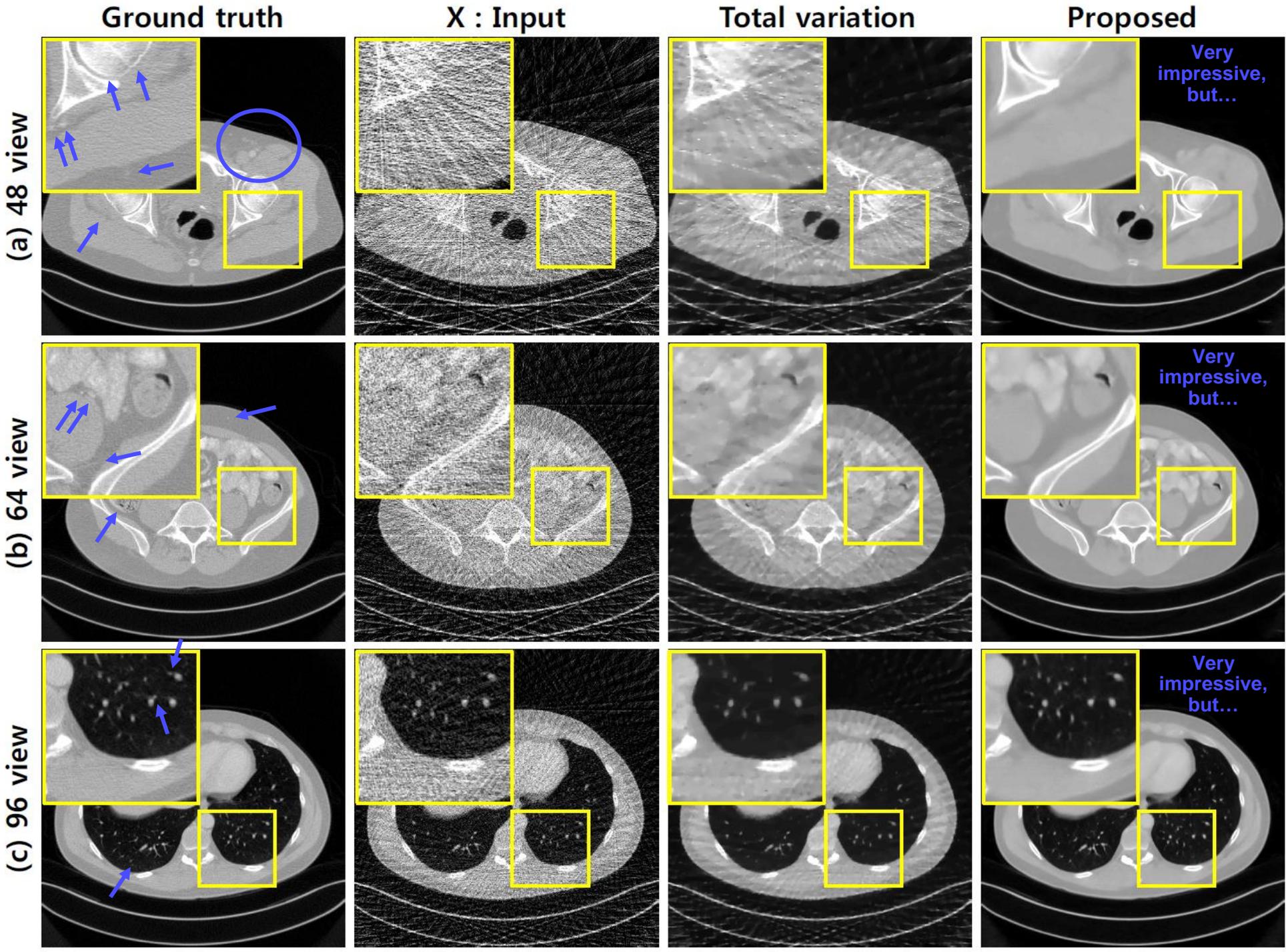


- followed by segmentation into tissue classes
- followed by forward projection of the CNN prior and replacement of metal areas of the original sinogram
- followed by reconstruction

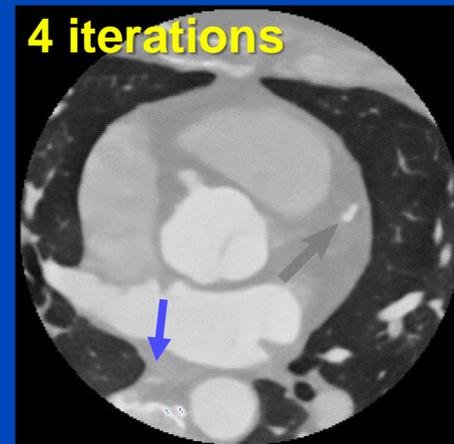
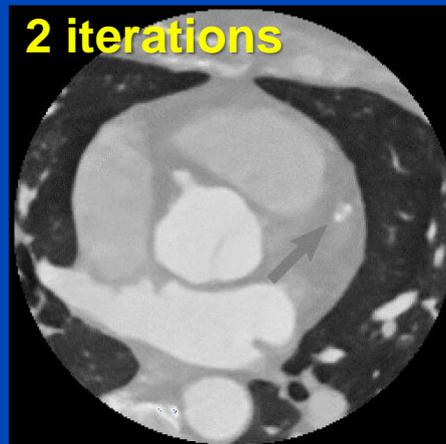
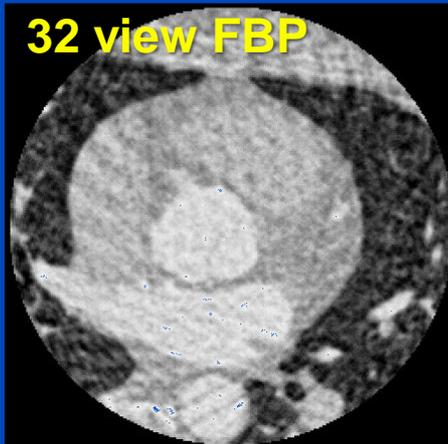
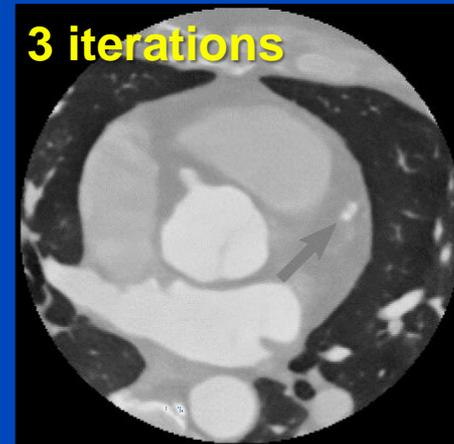
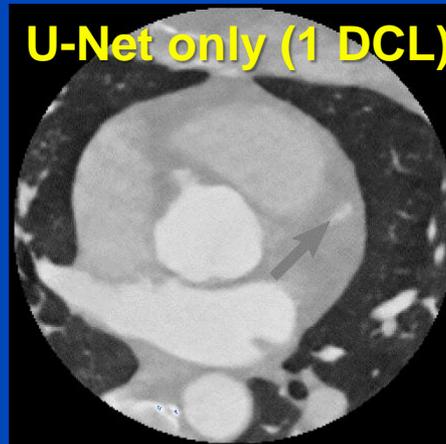
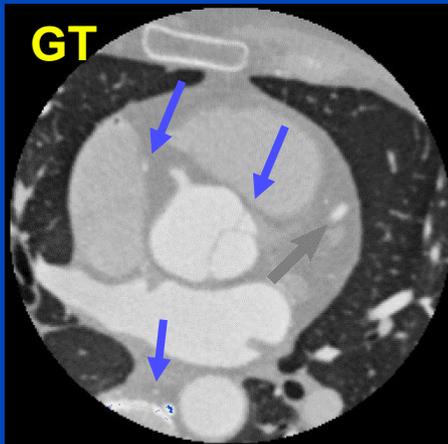
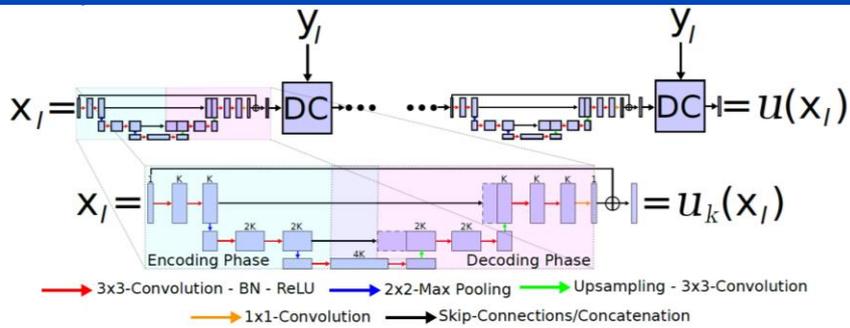


# Sparse View Restoration Example





# Sparse CT Recon with Data Consistency Layers (DCLs)

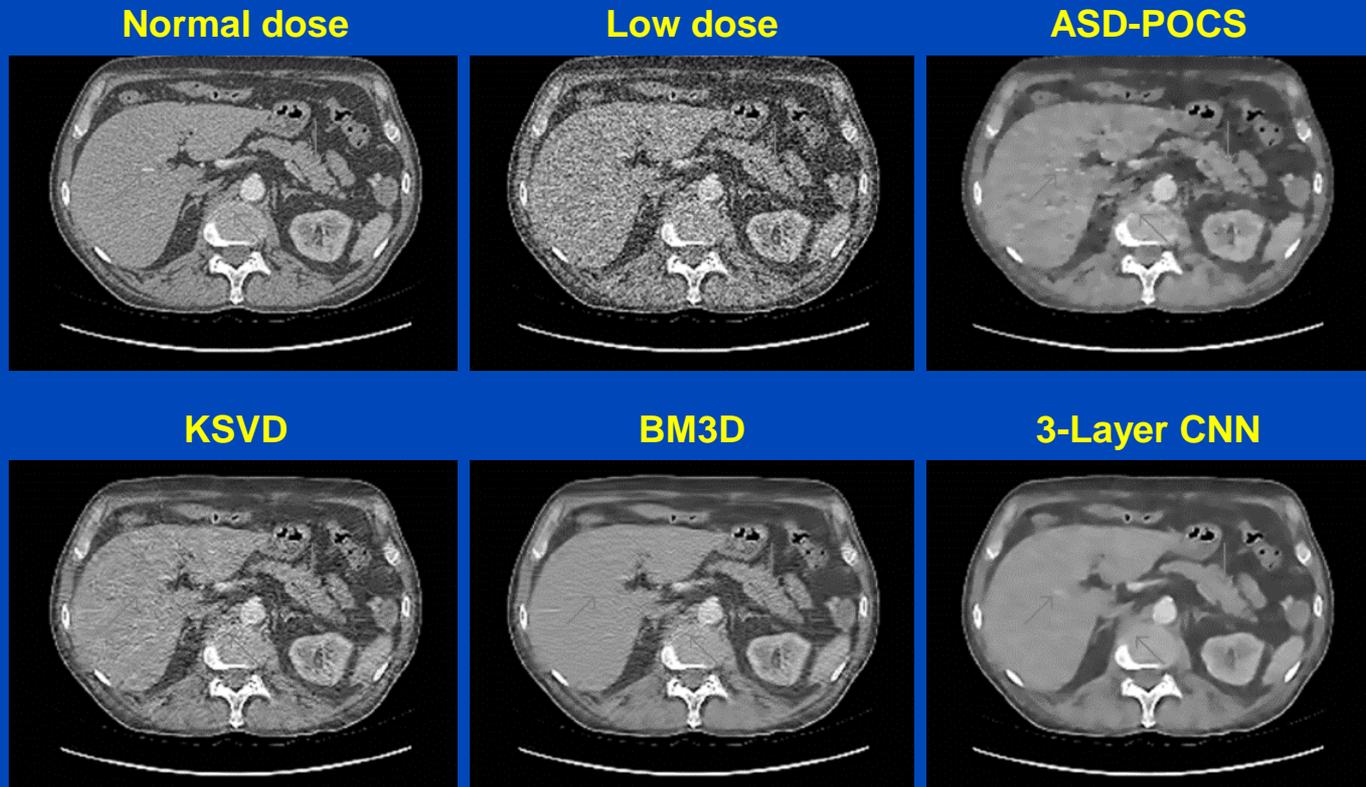


# Part 2:

# Noise Reduction

# Noise Removal Example 1

- 3-layer CNN uses low dose and corresponding normal dose image patches for training



# Noise Removal Example 2

- Task: Reduce noise from low dose CT images.
- A conditional generative adversarial networks (GAN) is used

- **Generator  $G$ :**

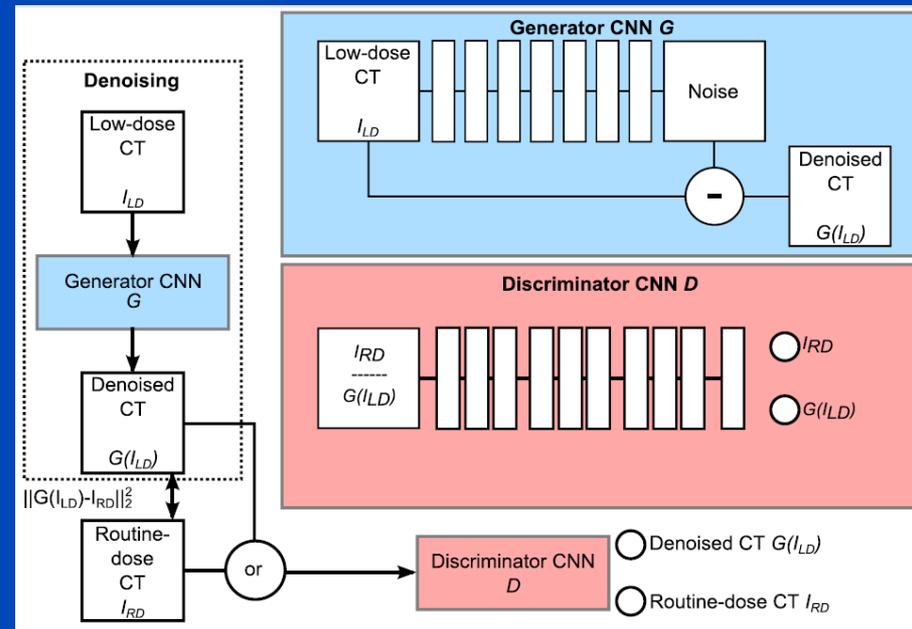
- 3D CNN that operates on small cardiac CT sub volumes
- Seven  $3 \times 3 \times 3$  convolutional layers yielding a receptive field of  $15 \times 15 \times 15$  voxels for each destination voxel
- Depths (features) from 32 to 128
- Batch norm only in the hidden layers
- Subtracting skip connection

- **Discriminator  $D$ :**

- Sees either routine dose image or a generator-denoised low dose image
- Two  $3 \times 3 \times 3$  layers followed by several  $3 \times 3$  layers with varying strides
- Feedback from  $D$  prevents smoothing.

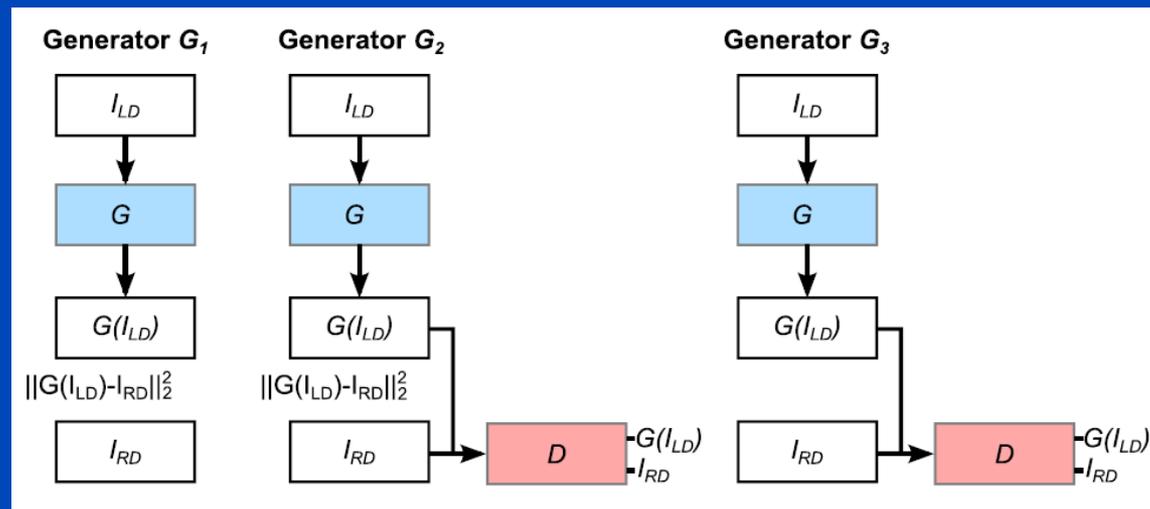
- **Training:**

- Unenhanced (why?) patient data acquired with Philips Brilliance iCT 256 at 120 kV.
- Two scans (why?) per patient, one with 0.2 mSv and one with 0.9 mSv effective dose.



# Noise Removal Example 2

- $G_1$  and  $G_2$  include supervised learning and thus perform only with phantom measurements.
- $G_3$  is unsupervised.
- $G_3$  is said to generate images with a more similar appearance to the routine-dose CT. Feedback from the discriminator  $D$  prevents smoothing the image.



# Noise Removal Example 2



Low dose image (0.2 mSv)

# Noise Removal Example 2



iDose level 3 reconstruction (0.2 mSv)

# Noise Removal Example 2



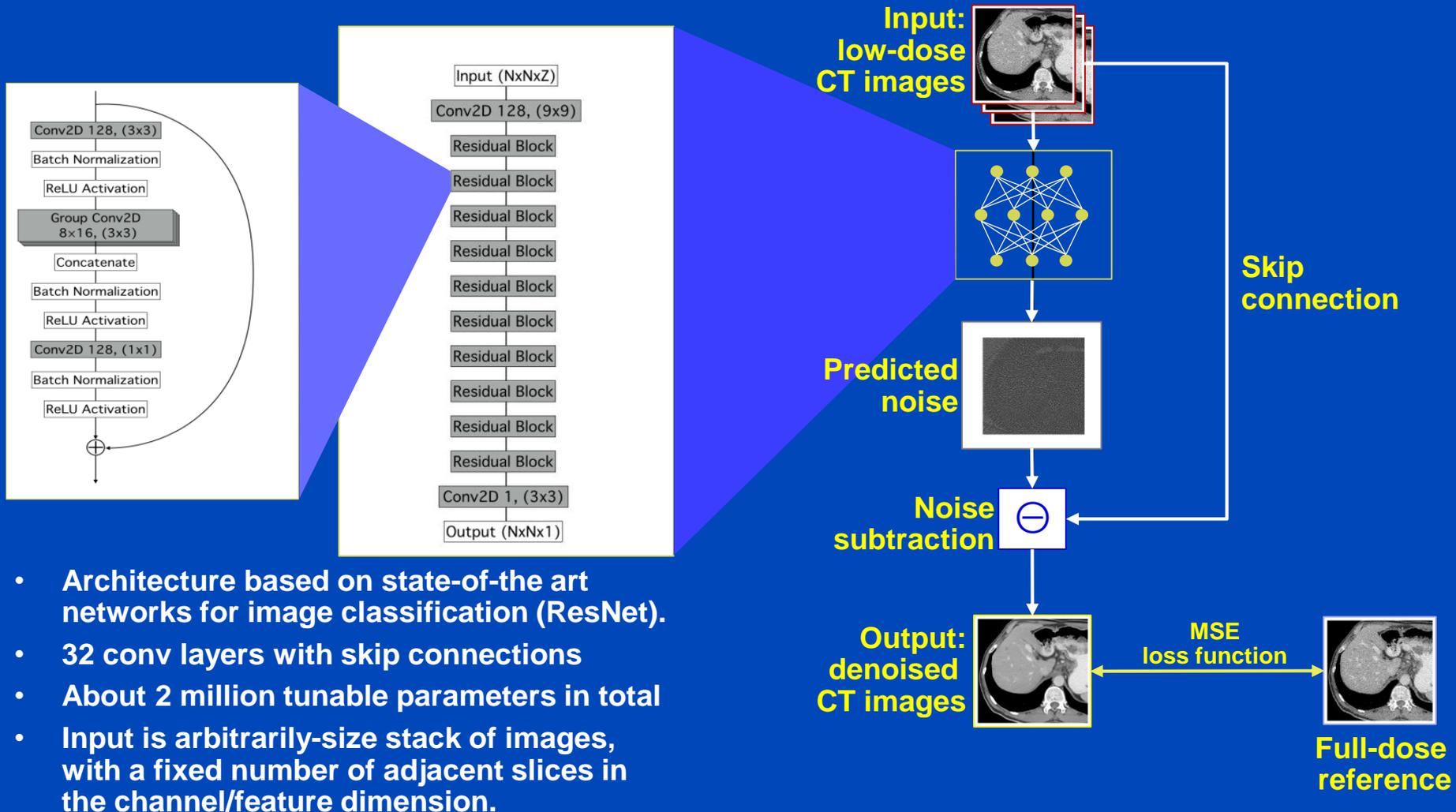
Denoised low dose image (0.2 mSv)

# Noise Removal Example 2

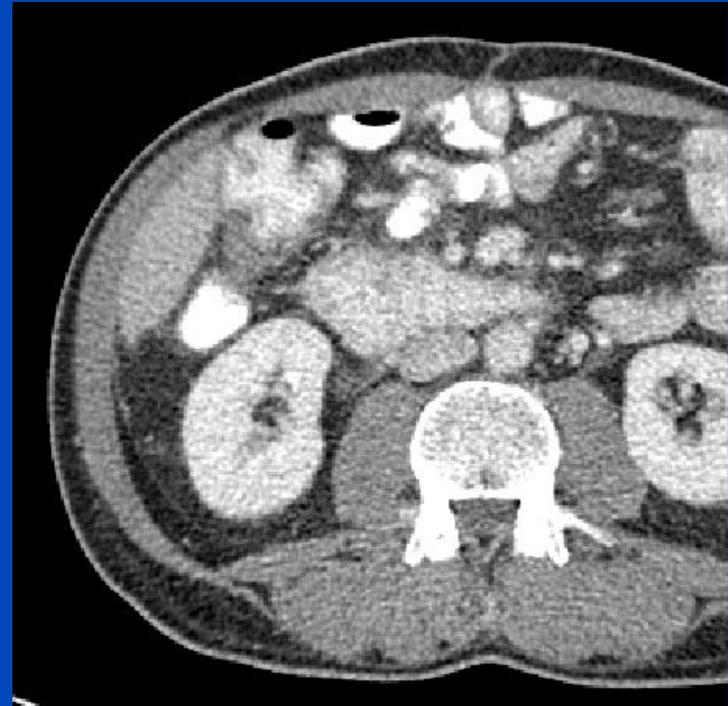


Normal dose image (0.9 mSv)

# Noise Removal Example 3

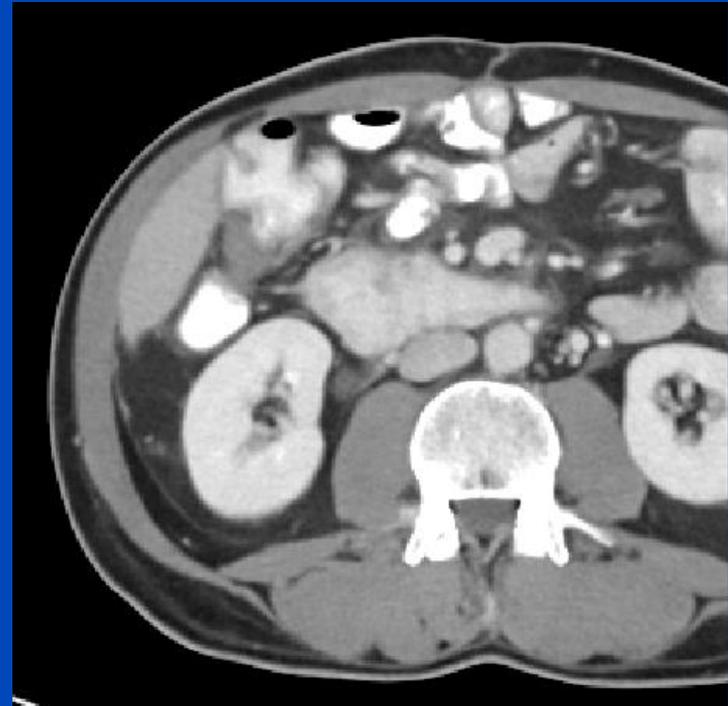
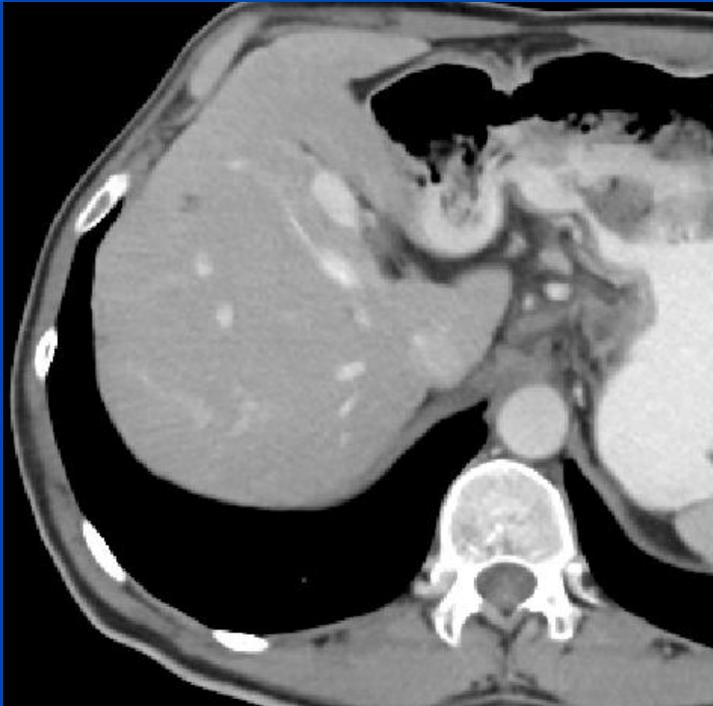


# Noise Removal Example 3



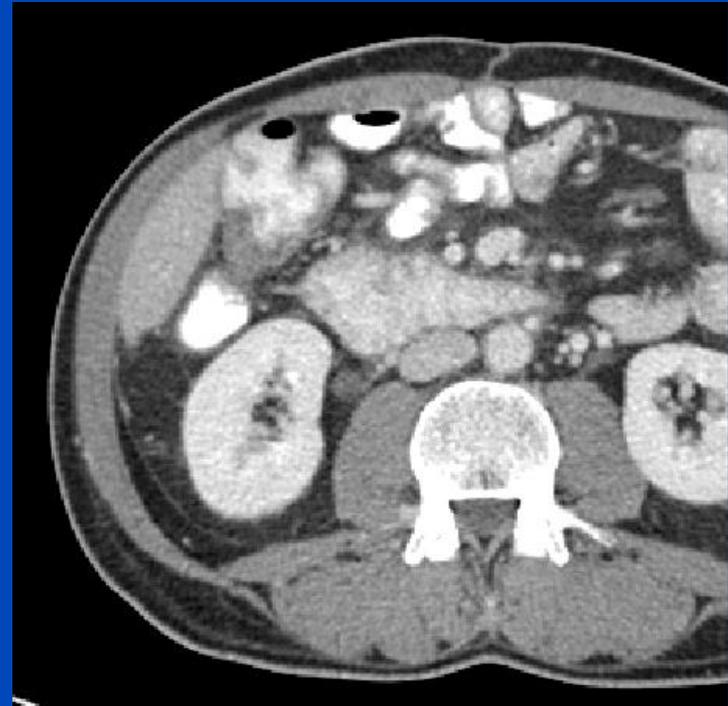
Low dose images (1/4 of full dose)

# Noise Removal Example 3



Denoised low dose

# Noise Removal Example 3



Full dose

# Noise Removal Example 3



Denoised full dose

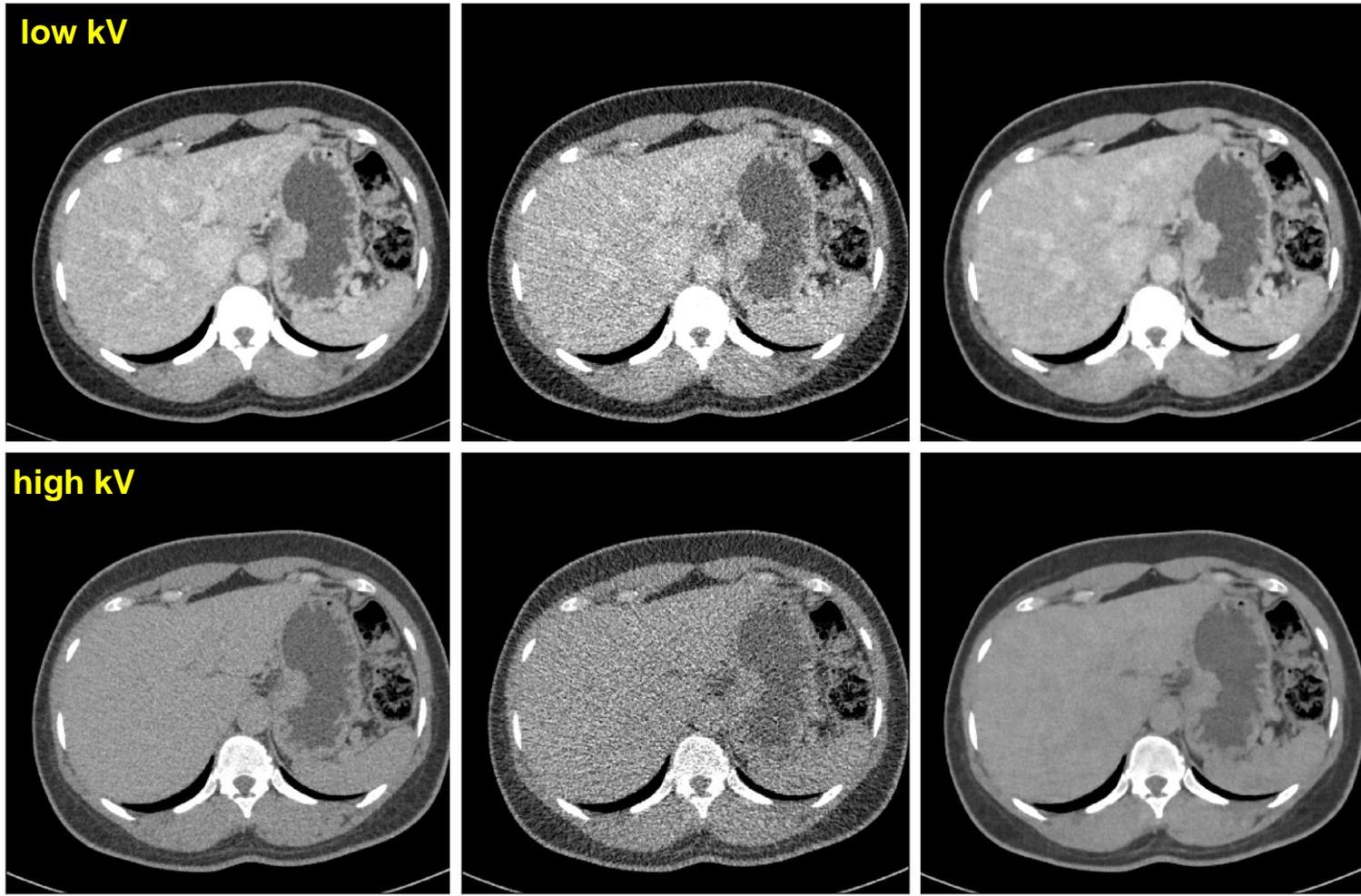
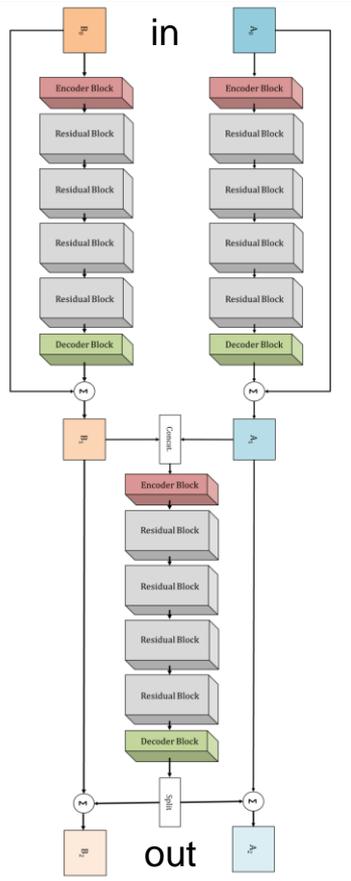
# ... and its Extension to DECT

low kV high kV

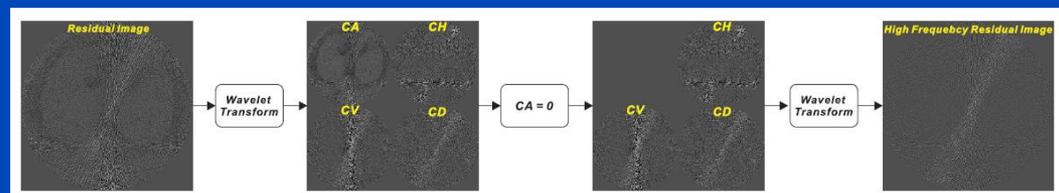
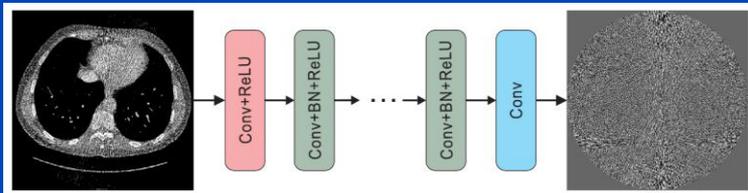
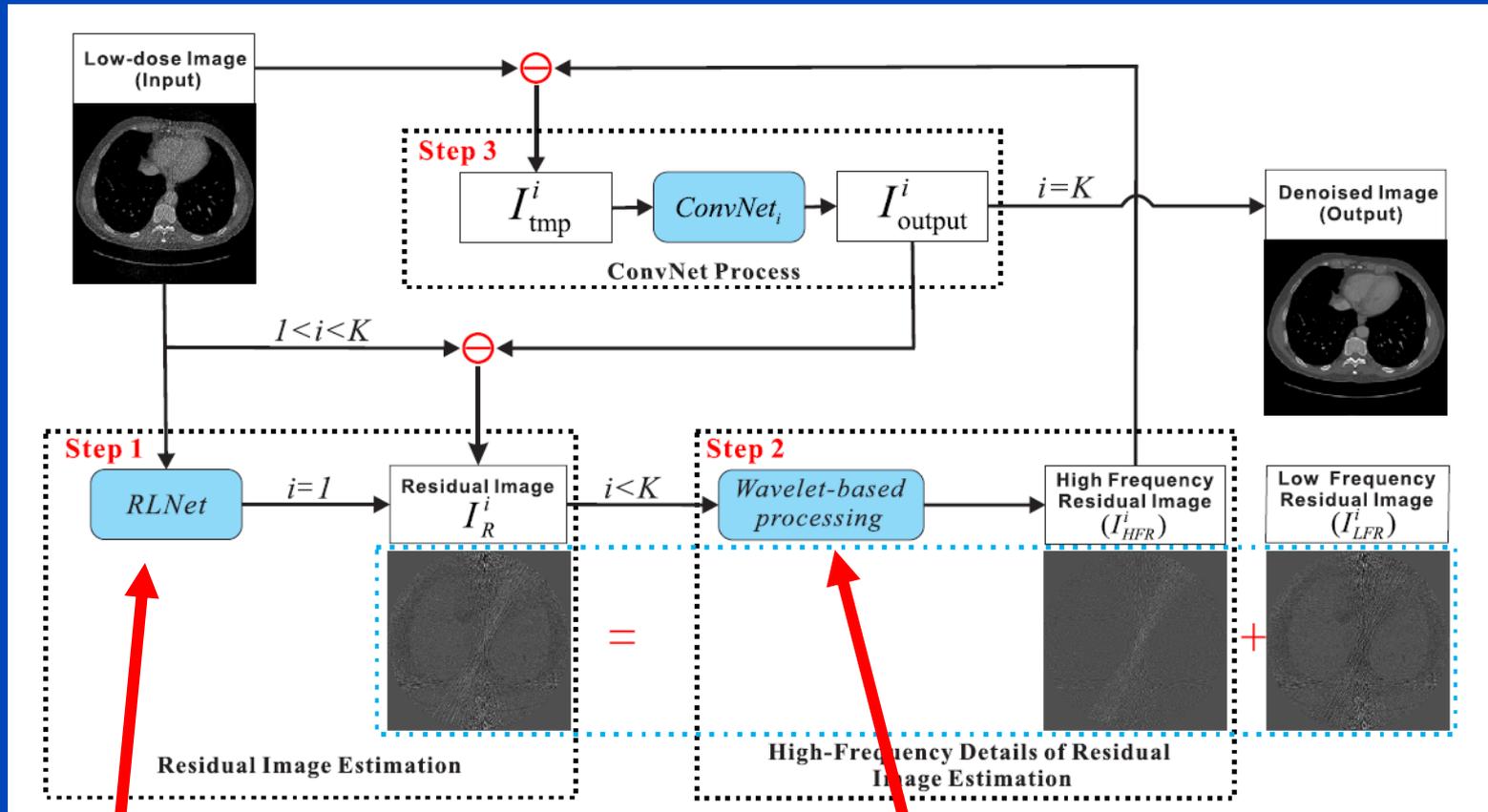
Full dose

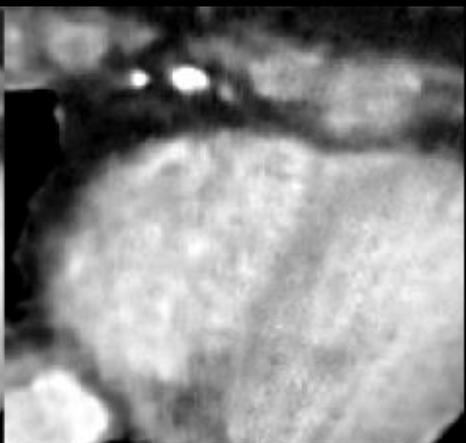
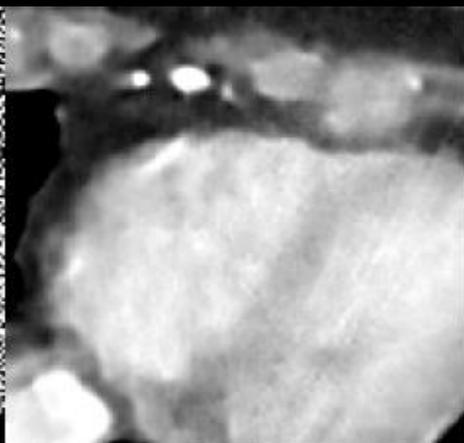
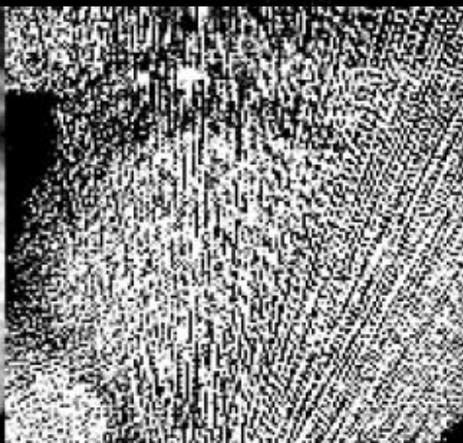
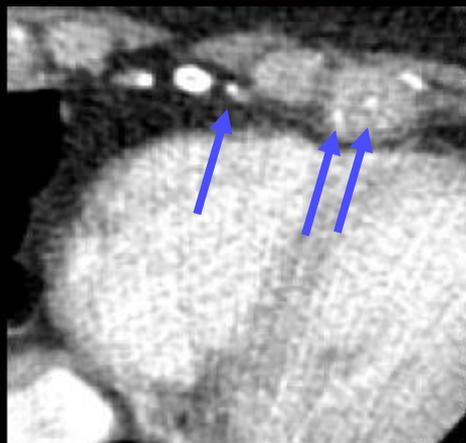
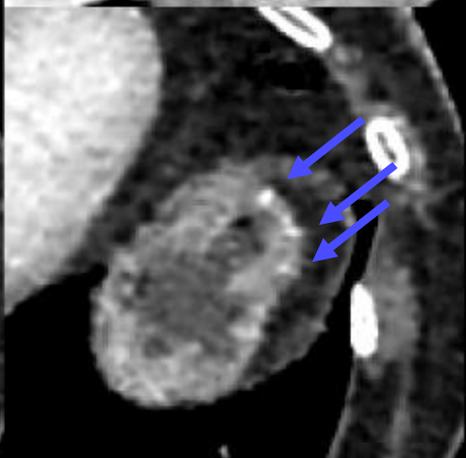
Low dose  
(1/4 of full dose)

Denoised low dose



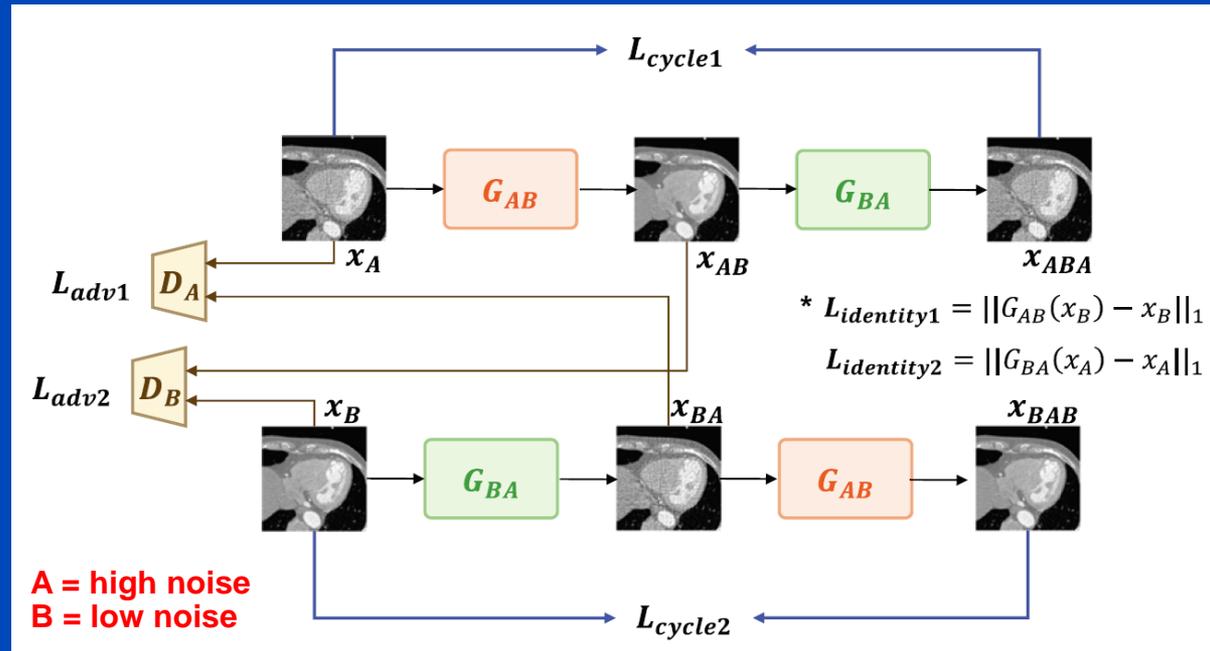
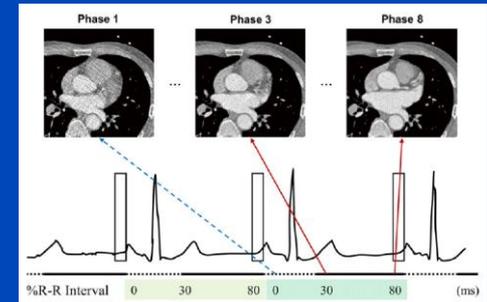
# Noise Removal Example 4



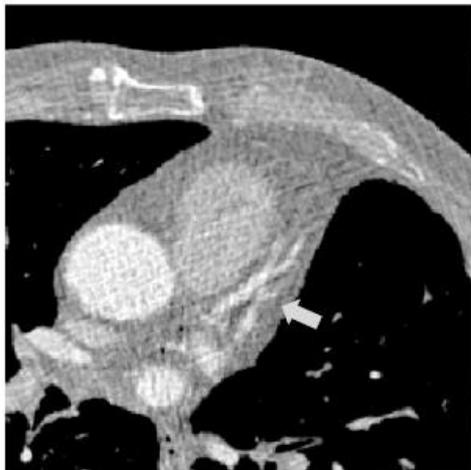
**FBP(200 mAs)****FBP(10 mAs)****IRLNet(10 mAs, T-Net)****IRLNet(10 mAs, A-Net)****ROI 1****ROI 2****ROI 3**

# Noise Removal Example 5

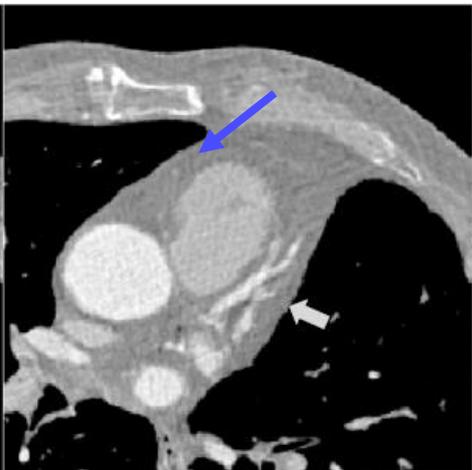
- ECG-based TCM yields cardiac phases with high noise.
- Train a cycle GAN that learns from the low noise phases to remove noise in the high noise phases.
- 50 patient cases used for training.
- Nice results!



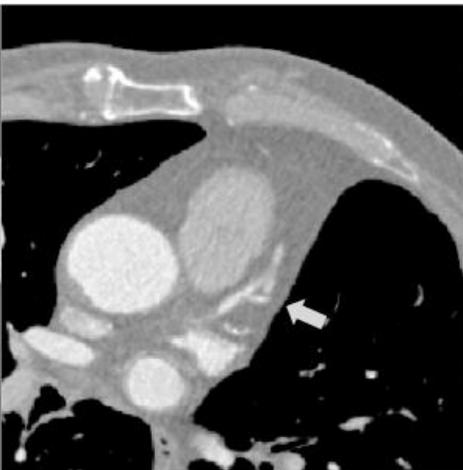
Input: Phase 1



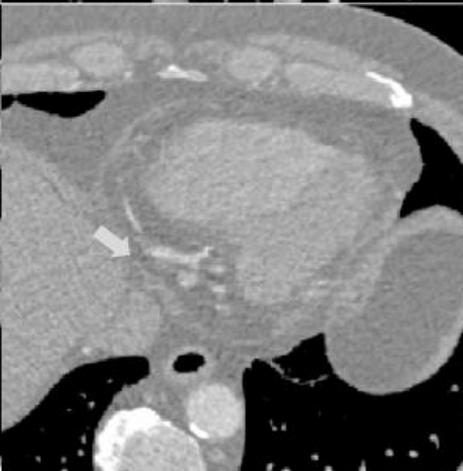
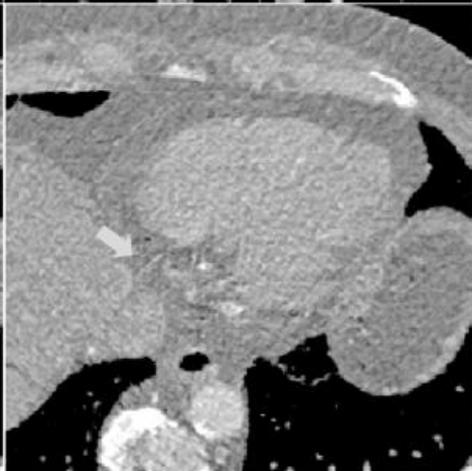
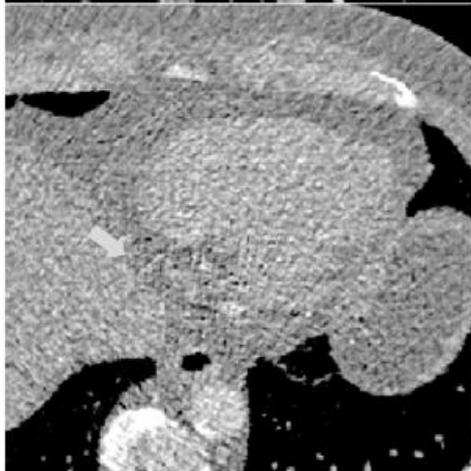
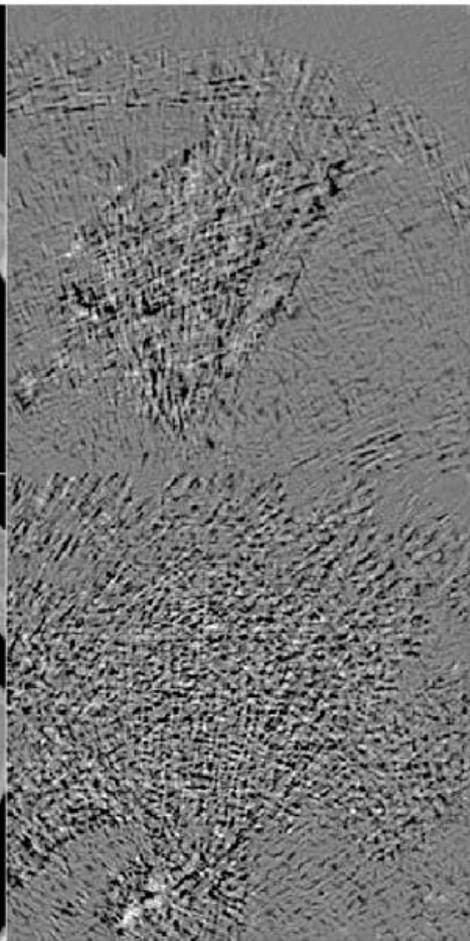
Result



Target: Phase 8



Input - Result



Input: Phase 1

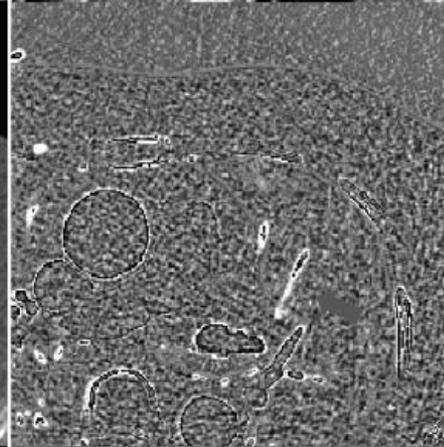
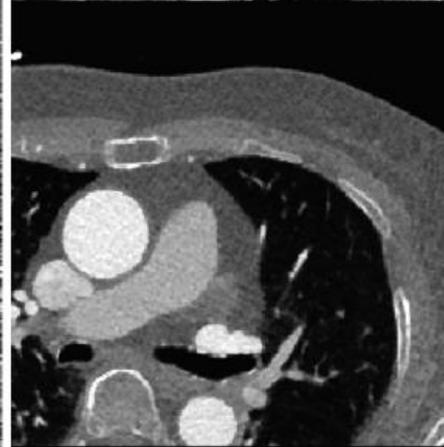
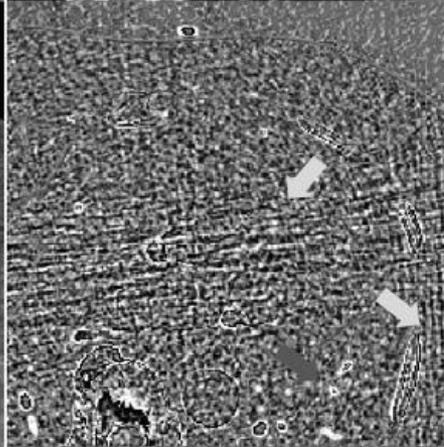
Target: Phase 8

Input: Phase 1

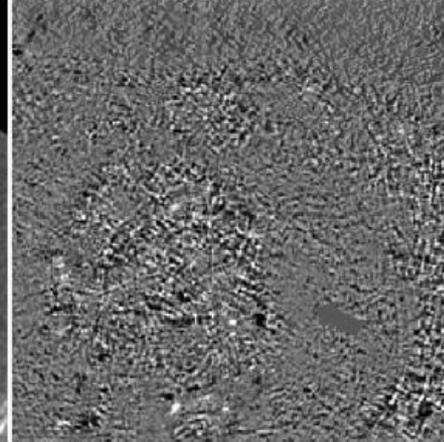
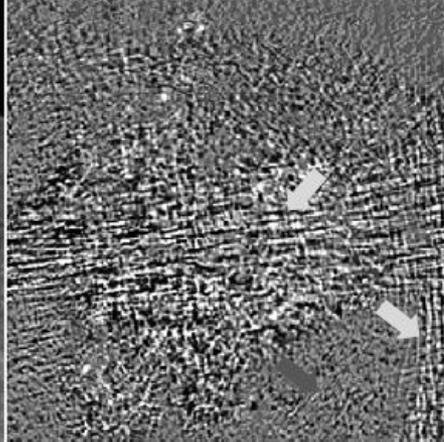
Target: Phase 8



ADMIRE



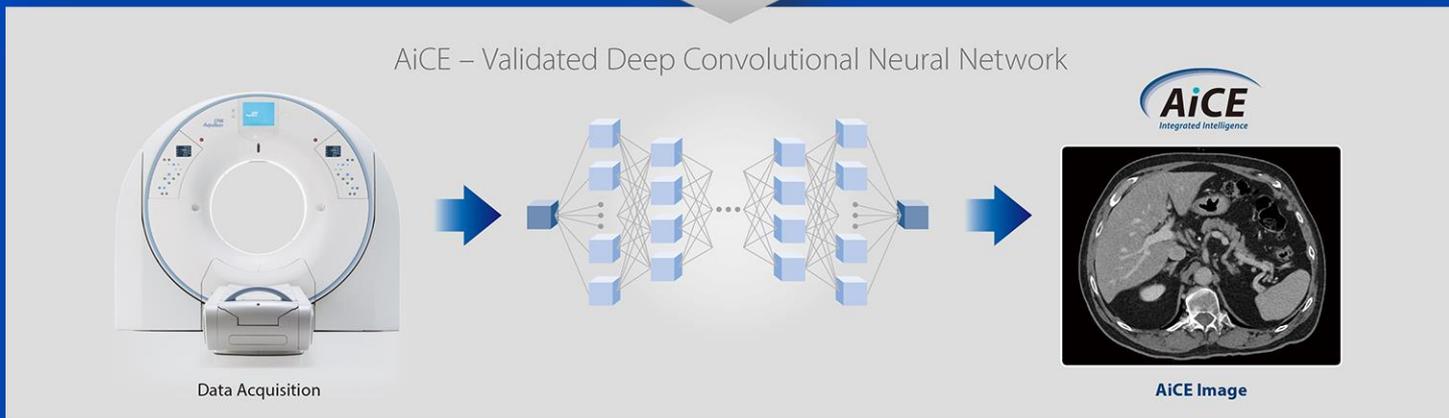
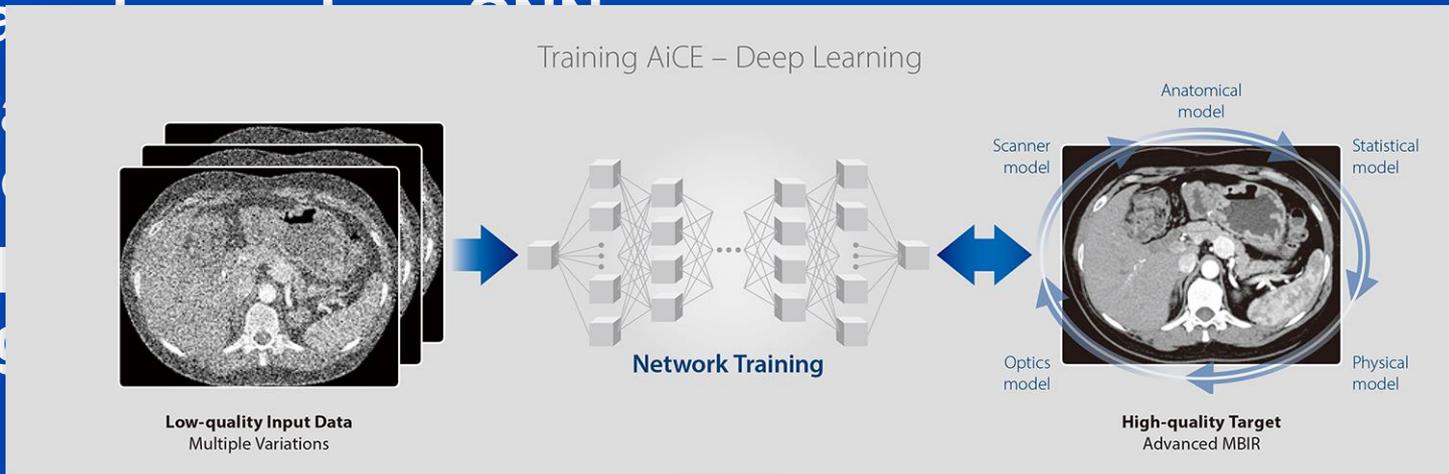
Proposed



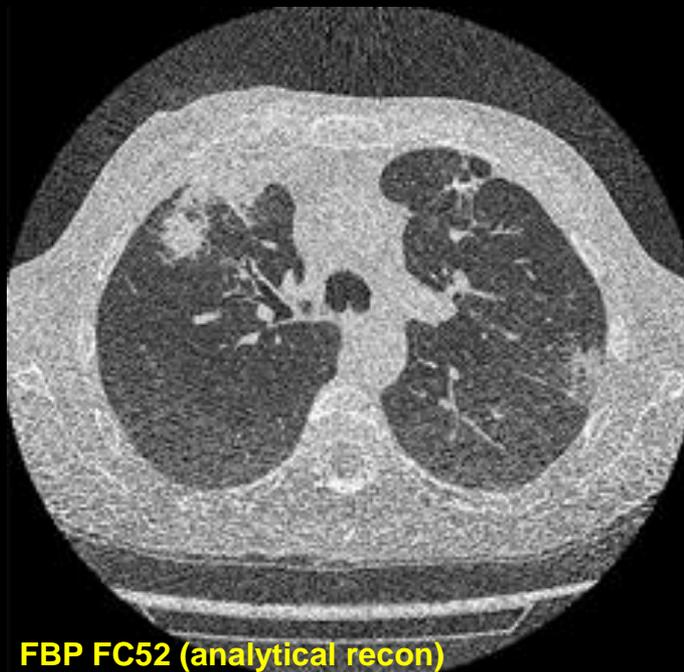
# Noise Removal Example 6

## Canon's AiCE

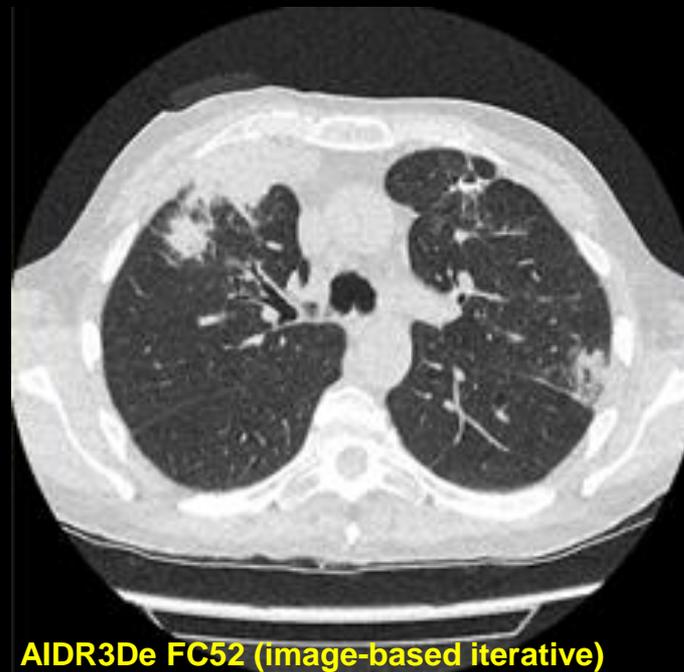
- Background: Deep Learning
- Training process
- Filter high quality



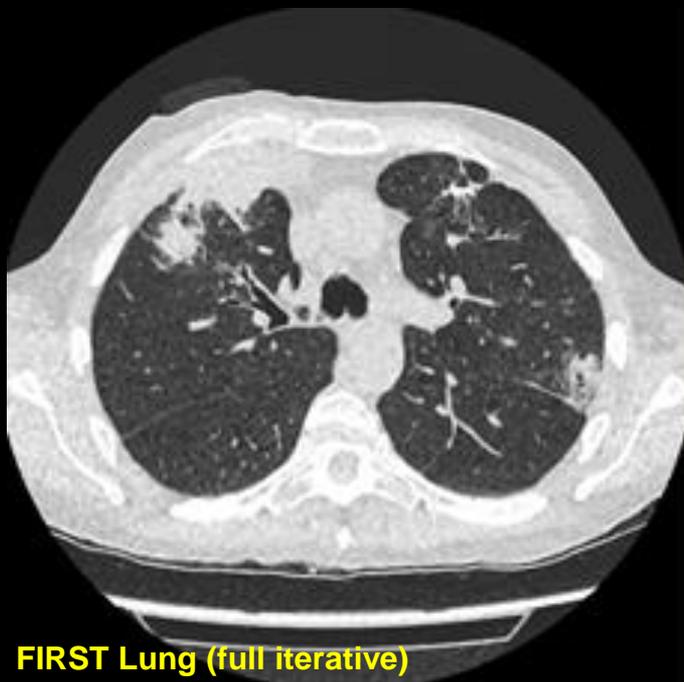
U = 100 kV  
CTDI = 0.6 mGy  
DLP = 24.7 mGy·cm  
D<sub>eff</sub> = 0.35 mSv



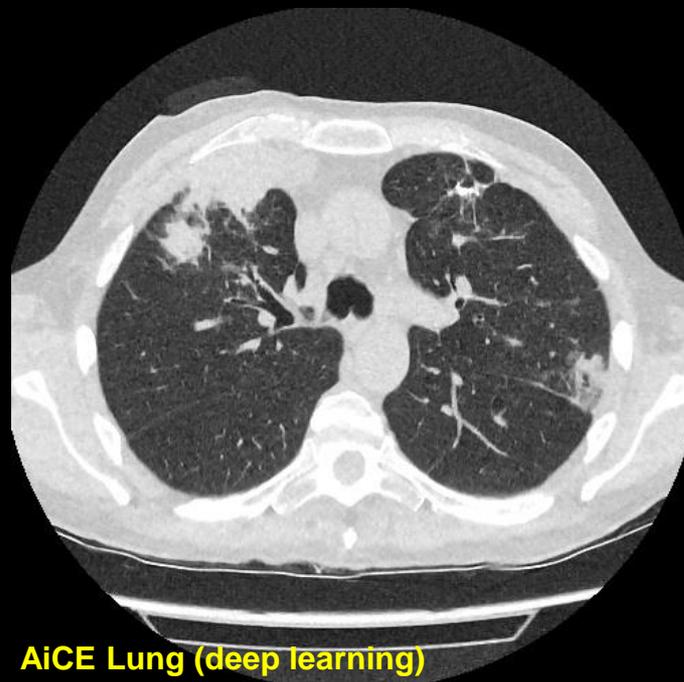
**FBP FC52 (analytical recon)**



**AIDR3De FC52 (image-based iterative)**



**FIRST Lung (full iterative)**



**AiCE Lung (deep learning)**

# Noise Removal Example 7

## GE's True Fidelity

- Based on a deep CNN
- Trained to restore low-dose CT data to match the properties of Veo, the model-based IR of GE.
- No information can be obtained in how the training is conducted for the product implementation.

### 2.5D DEEP LEARNING FOR CT IMAGE RECONSTRUCTION USING A MULTI-GPU IMPLEMENTATION

Amirkoushyar Ziabari\*, Dong Hye Ye \* †, Somesh Srivastava ‡, Ken D. Sauer ⊕  
Jean-Baptiste Thibault ‡, Charles A. Bouman\*

\* Electrical and Computer Engineering at Purdue University

† Electrical and Computer Engineering at Marquett University

‡ GE Healthcare

⊕ Electrical Engineering at University of Notre Dame

#### ABSTRACT

While Model Based Iterative Reconstruction (MBIR) of CT scans has been shown to have better image quality than Filtered Back Projection (FBP), its use has been limited by its high computational cost. More recently, deep convolutional neural networks (CNN) have shown great promise in both denoising and reconstruction applications. In this research, we propose a fast reconstruction algorithm, which we call Deep Learning MBIR (DL-MBIR).

streaking artifacts caused by sparse projection views in CT images [8]. More recently, Ye, et al. [9] developed method for incorporating CNN denoisers into MBIR reconstruction as advanced prior models using the Plug-and-Play framework [10, 11].

In this paper, we propose a fast reconstruction algorithm, which we call Deep Learning MBIR (DL-MBIR), for approximately achieving the improved quality of MBIR using a deep residual neural network. The DL-MBIR method is trained to

# No Low Noise Images Required to Train Denoising Networks!

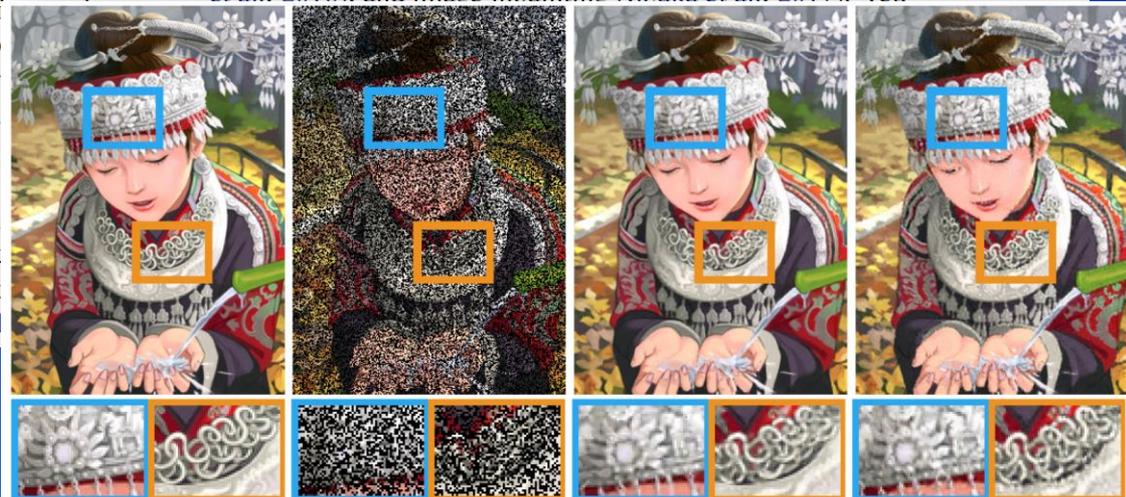
## Noise2Noise: Learning Image Restoration without Clean Data

Jaakko Lehtinen<sup>1,2</sup> Jacob Munkberg<sup>1</sup> Jon Hasselgren<sup>1</sup> Samuli Laine<sup>1</sup> Tero Karras<sup>1</sup> Miika Aittala<sup>3</sup> Timo Aila<sup>1</sup>

### Abstract

We apply basic statistical reasoning to signal reconstruction by machine learning – learning to map corrupted observations to clean signals – with a simple and powerful conclusion: it is possible to learn to restore images by only looking at corrupted examples, at performance at times exceeding training using clean data, without explicit image priors or likelihood model of corruption. In practice, we show that our model learns photographic noise removal on synthetic Monte Carlo images, and denoising of undersampled MRI scans – all compared by different processes – based on noisy d

renderings of a synthetic scene, etc. Significant advances have been reported in several applications, including Gaussian denoising, de-JPEG, text removal (Mao et al., 2016), super-resolution (Ledig et al., 2017), colorization (Zhang et al., 2016), and image inpainting (Iizuka et al., 2017). Yet,



Ground truth

Input

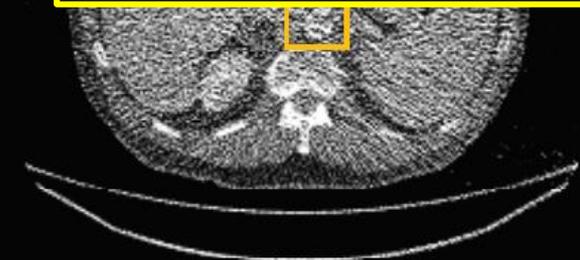
Our

Comparison

# No Low Noise Images Required to Train Denoising Networks!

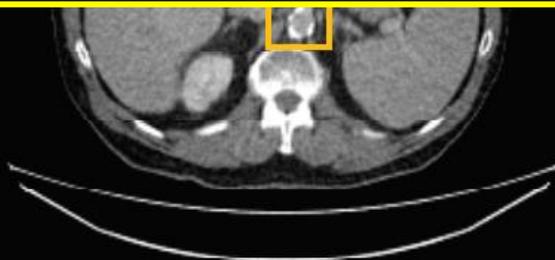
- Estimation can be regarded as ML estimation by interpreting the loss function as the negative log likelihood.
- On expectation, the estimate remains unchanged if we replace the targets with random numbers whose expectations match the targets.
- Input-conditioned target distributions  $p(y|x)$  can be replaced with arbitrary distributions that have the same conditional expected values.
- Consequently, we may corrupt the training targets of a neural network with zero-mean noise without changing what the network learns.
- Useful for image restoration tasks where the expectation of the corrupted input data is the clean target that we seek to restore.
- **Denoising possible if at least two realizations of each image are available.**

# Noise Removal Example 8 (Training on Noisy CT Targets)



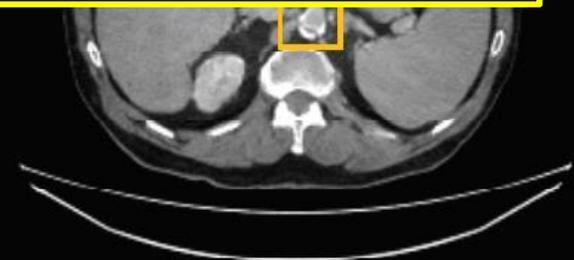
MAE:128.342, SNR:13.772, SSIM: 0.306

(a) Low-dose



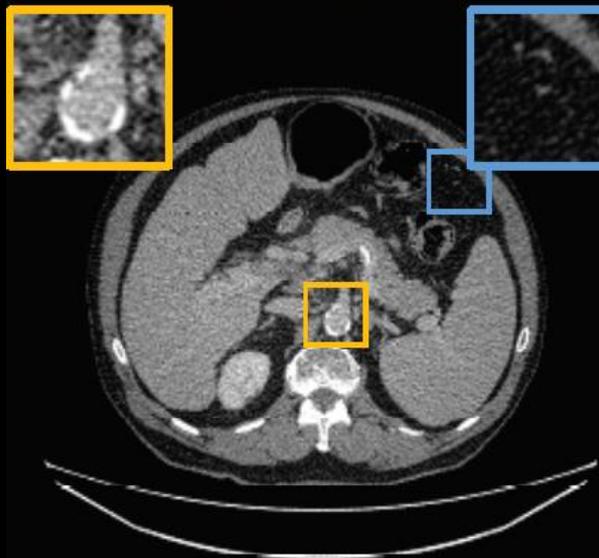
MAE:26.710, SNR:27.232, SSIM: 0.846

(b) Rawdata domain Ld2Ld

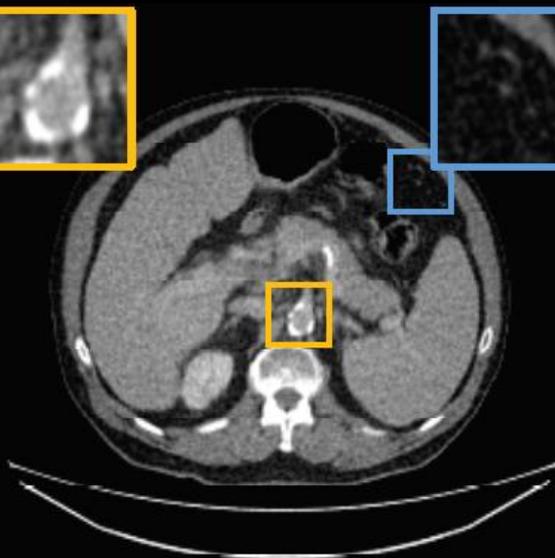


MAE:27.284, SNR:27.301, SSIM: 0.824

(c) Image-domain Ld2Ld

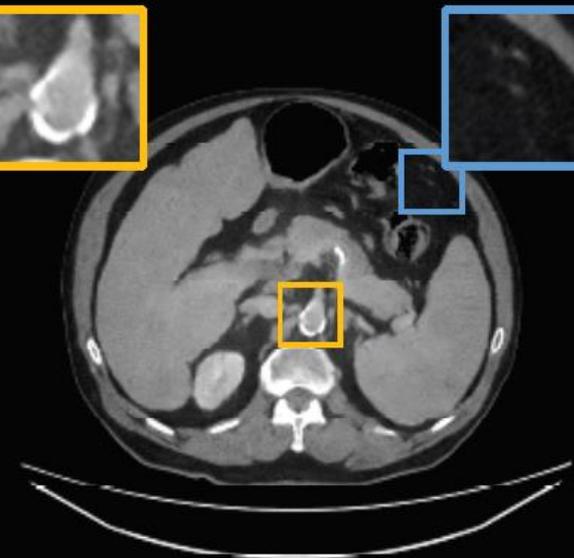


(d) Reference



MAE:26.595, SNR:27.281, SSIM: 0.839

(e) Rawdata domain Ld2Hd



MAE:26.135, SNR:27.725, SSIM: 0.845

(f) Image-domain Ld2Hd

## Part 3:

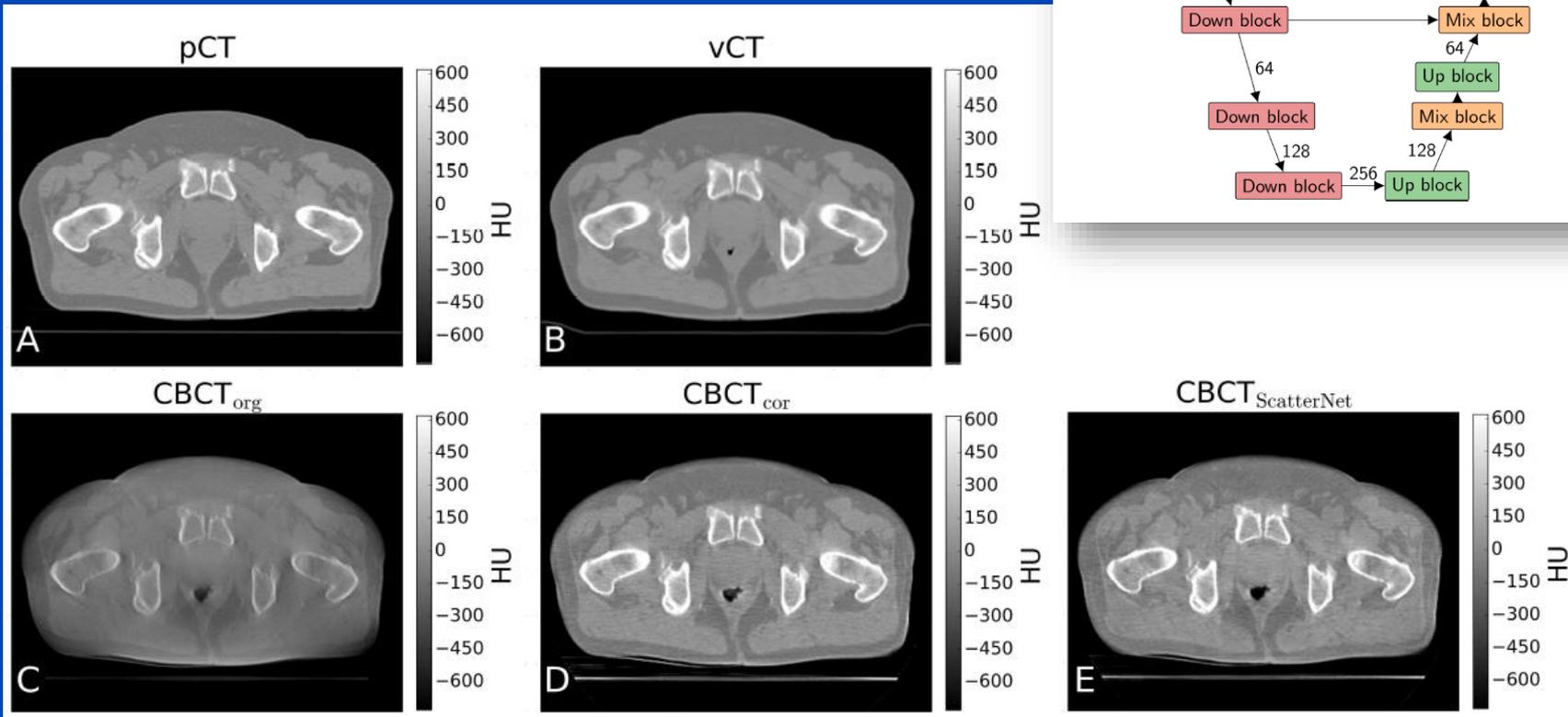
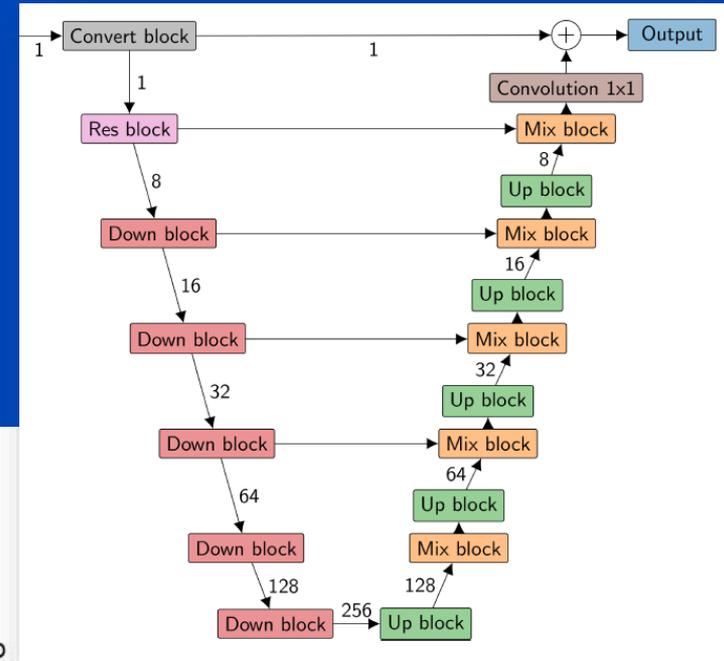
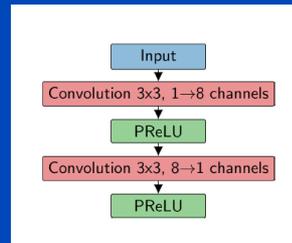
# Replacement of Lengthy Computations Fast Physics

# Empirical Shading Correction: ScatterNet

- Net to convert CBCT log (why?) rawdata into artifact-free data.
- Net architecture:
  - Small receptive field spectrum converter block adapts the attenuation values.
  - Residual U-Net then follows to account for scatter.
- Pixel-wise loss function comparing the corrected CBCT projections with those of the reference shading correction method.
- Reference shading correction method:
  - Use data from a clinical CT scan as an artifact-free prior.
  - Intensity domain frequency split between planning CT and CBCT:
    - » Deformably register planning CT onto CBCT and forward project and exponentiate to obtain “ideal” intensity data
    - » Scale CBCT intensities to match the prior CT intensities
    - » **Corrected intensities = LP(forward proj. CT)+HP(scaled uncorr. CBCT)**
- ScatterNet replaces the previous correction method and thus speeds up computation and does not make use of the planning CT.

# ScatterNet

## Spectrum converter block



# Deep Scatter Estimation

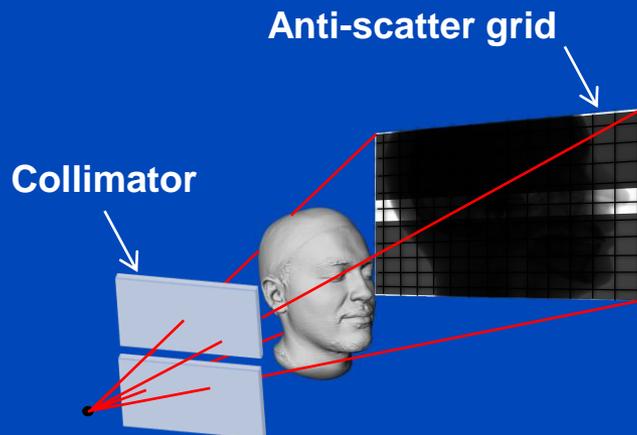
# Scatter Correction

## Scatter suppression

- Anti-scatter grids
- Collimators
- ...

## Scatter estimation

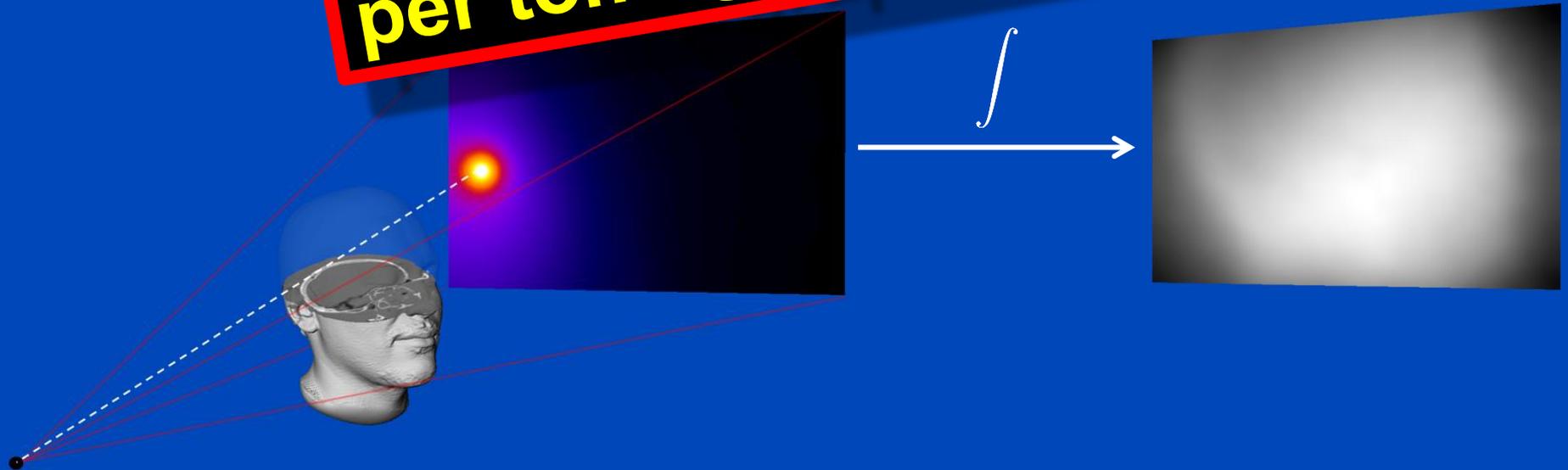
- Monte Carlo simulation
- Kernel-based approaches
- Boltzmann transport
- Primary modulation
- Beam blockers
- ...



# Monte Carlo Scatter Estimation

- Simulation of photon trajectories according to physical interaction probabilities.
- Simulating a large number of trajectories well approximates the complete scatter distribution

**1 to 10 hours  
per tomographic data set**



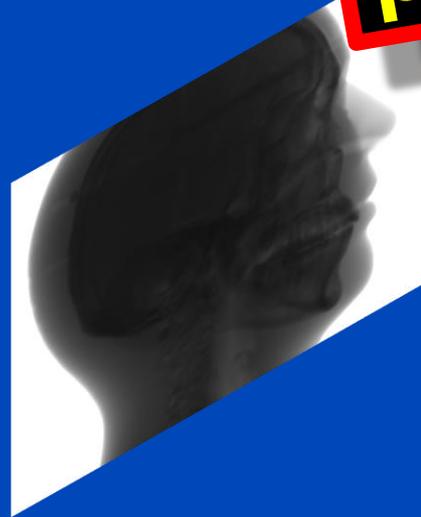
# Deep Scatter Estimation (DSE)

Train a deep convolutional neural network (CNN) to estimate scatter using a function of the input and projection data as input.

**0.1 to 1 minute  
per tomographic data set**

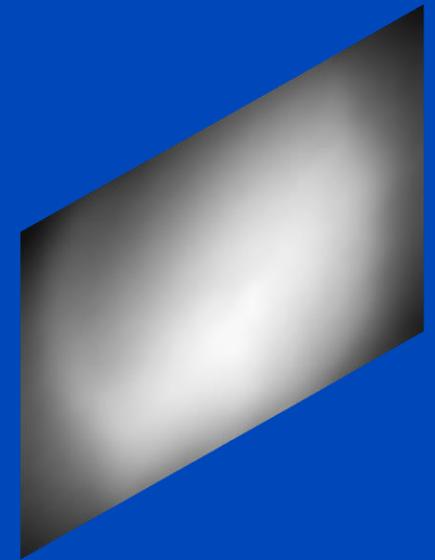
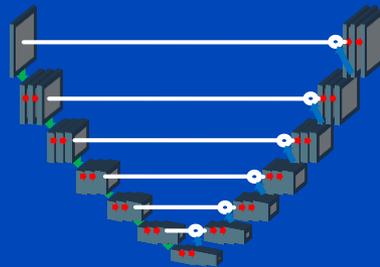
Input:  $T(p)$

Scatter estimate



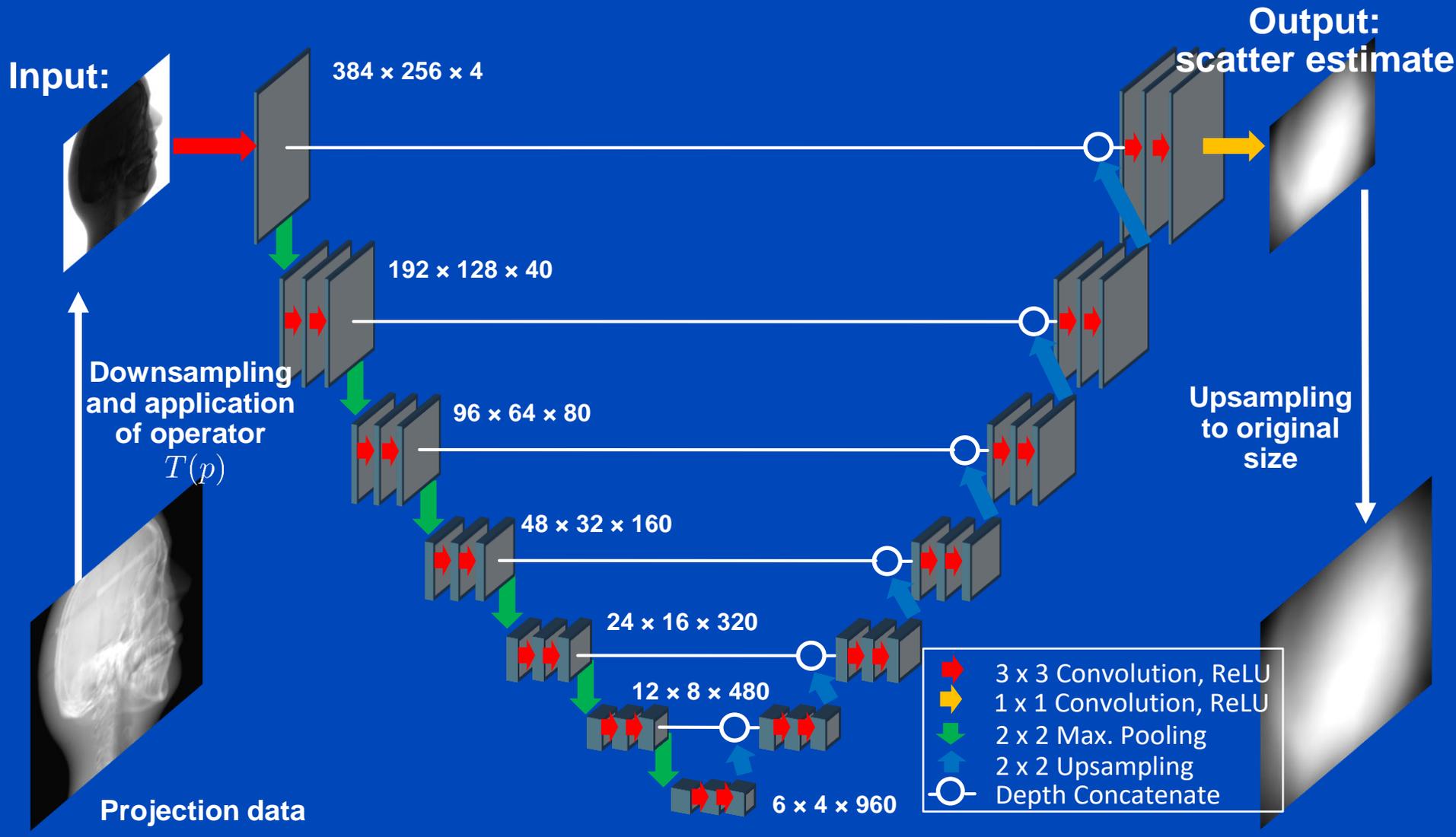
~~Monte Carlo~~

Convolutional neural network



# Deep Scatter Estimation

## Network architecture & scatter estimation framework



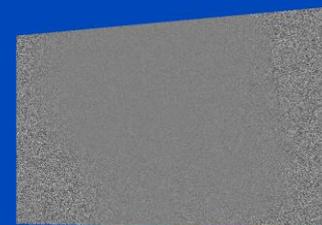
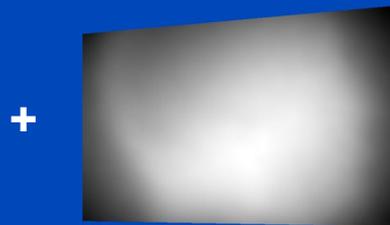
# Training the DSE Network

CBCT Setup

Primary intensity

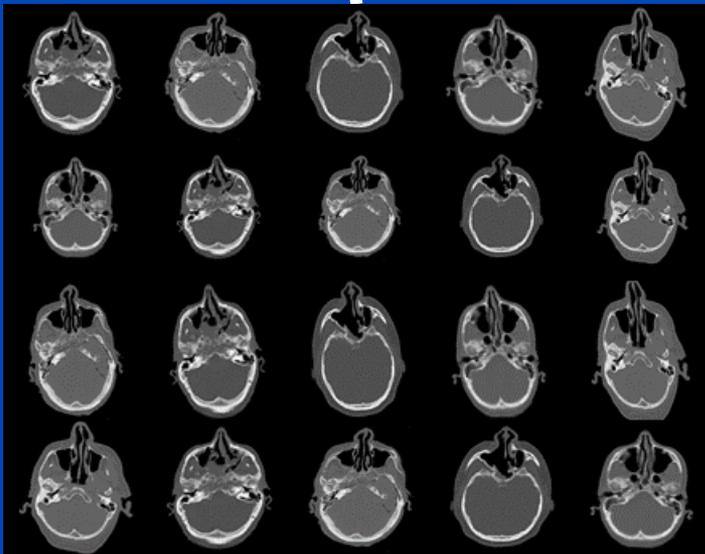
MC scatter simulation

Poisson noise



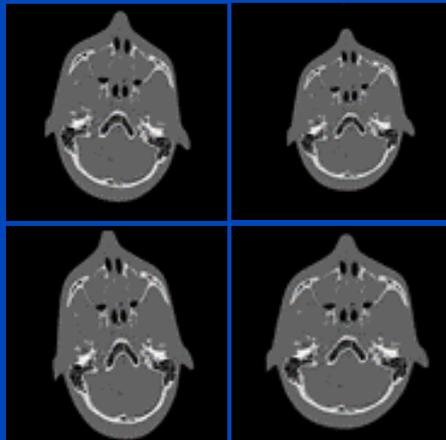
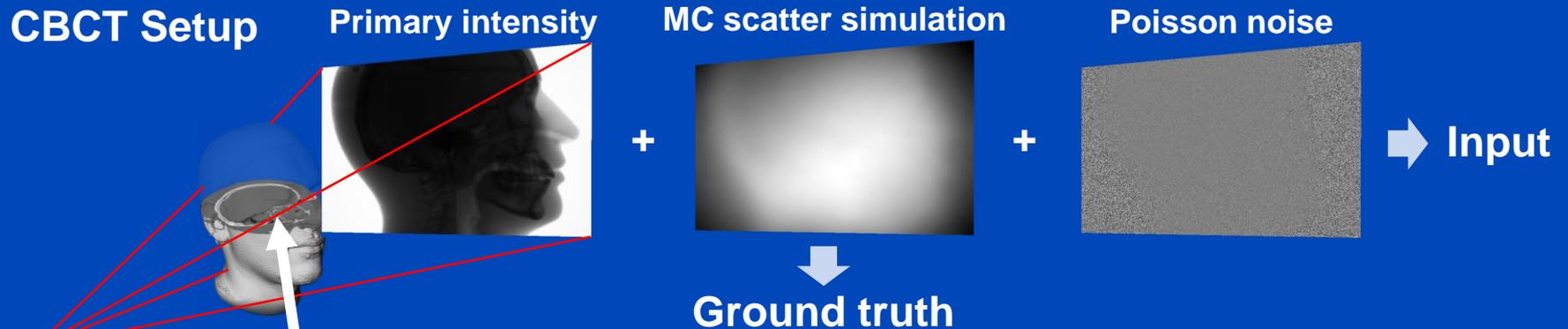
➔ Input

↓  
Desired output



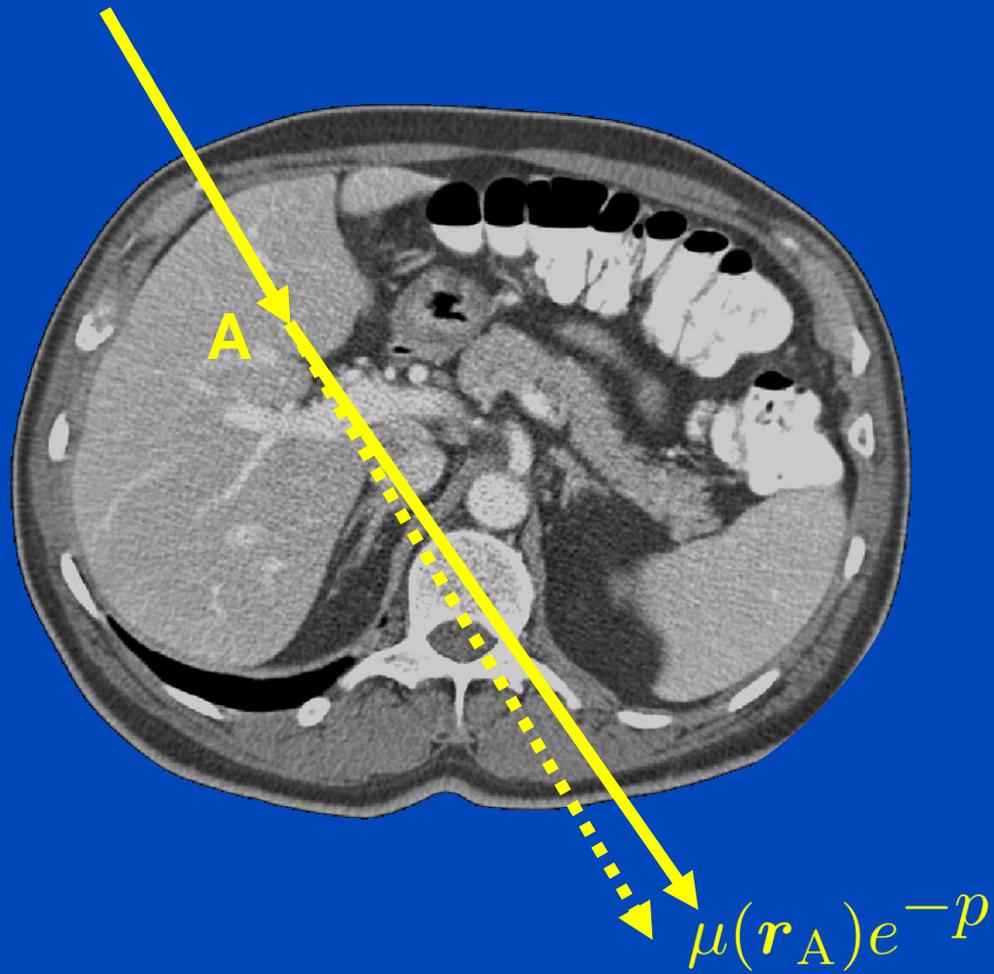
- Simulation of 6000 projections using different heads and acquisition parameters (80 kV, ..., 140 kV in steps of 20 kV).
- Splitting into 80% training and 20% validation data.
- Mean  $S/P = 0.9$
- 90<sup>th</sup> percentile  $S/P = 1.32$
- Training minimizes MSE pixel-wise loss on a GeForce GTX 1080 for 80 epochs.

# Testing of the DSE Network for Simulated Data (at 120 kV)

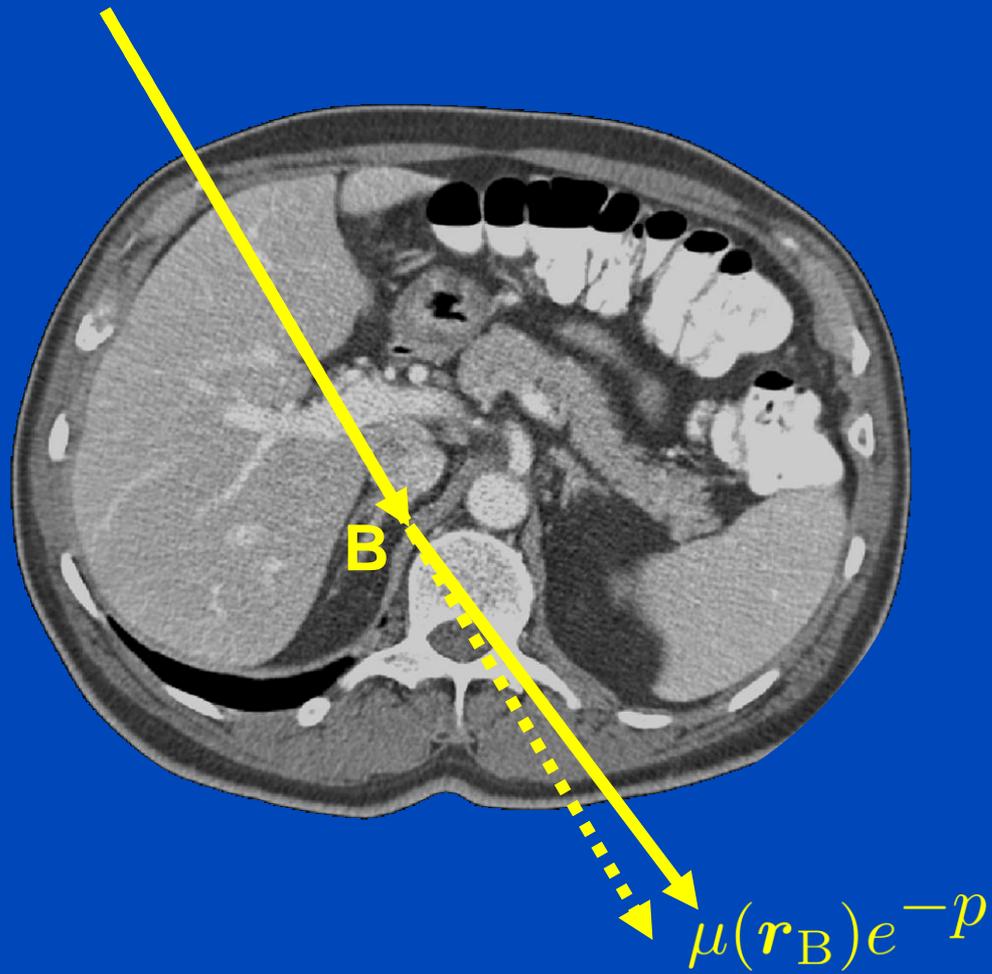


- Application of the DSE network to predict scatter for simulated data of a head (different from training data).

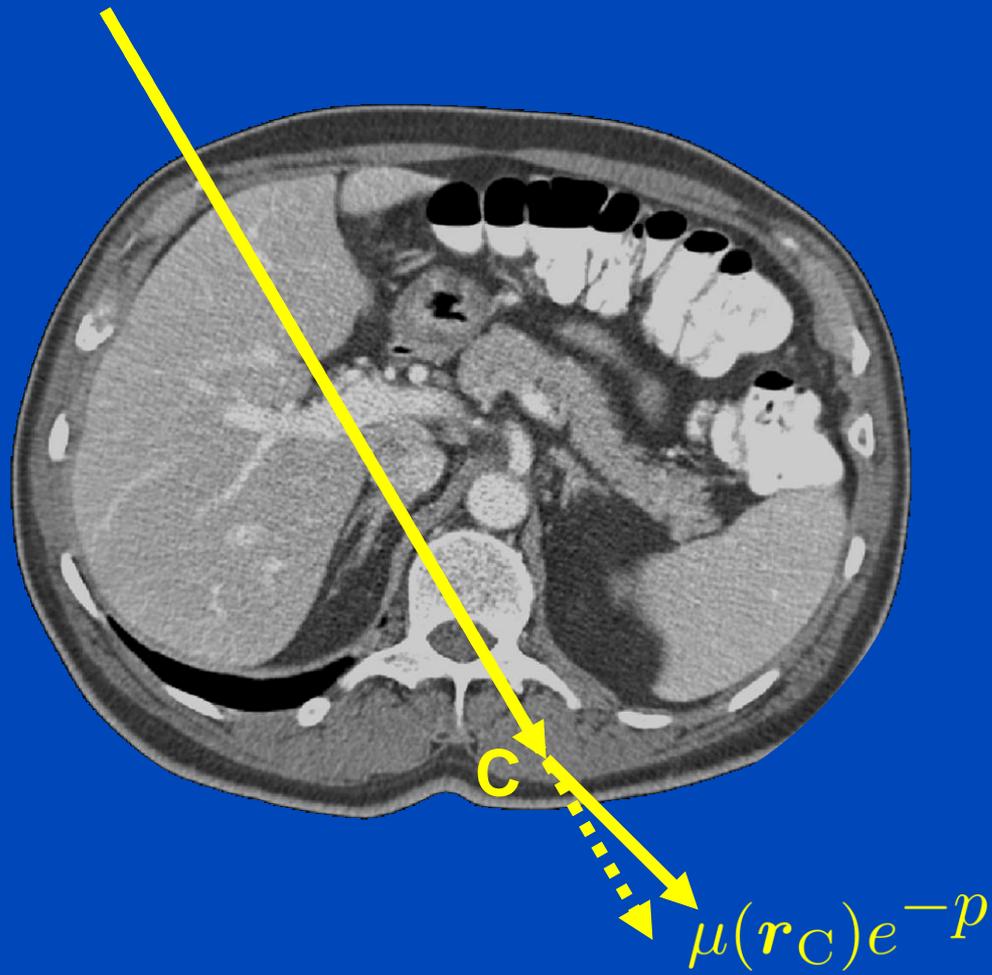
# PEP



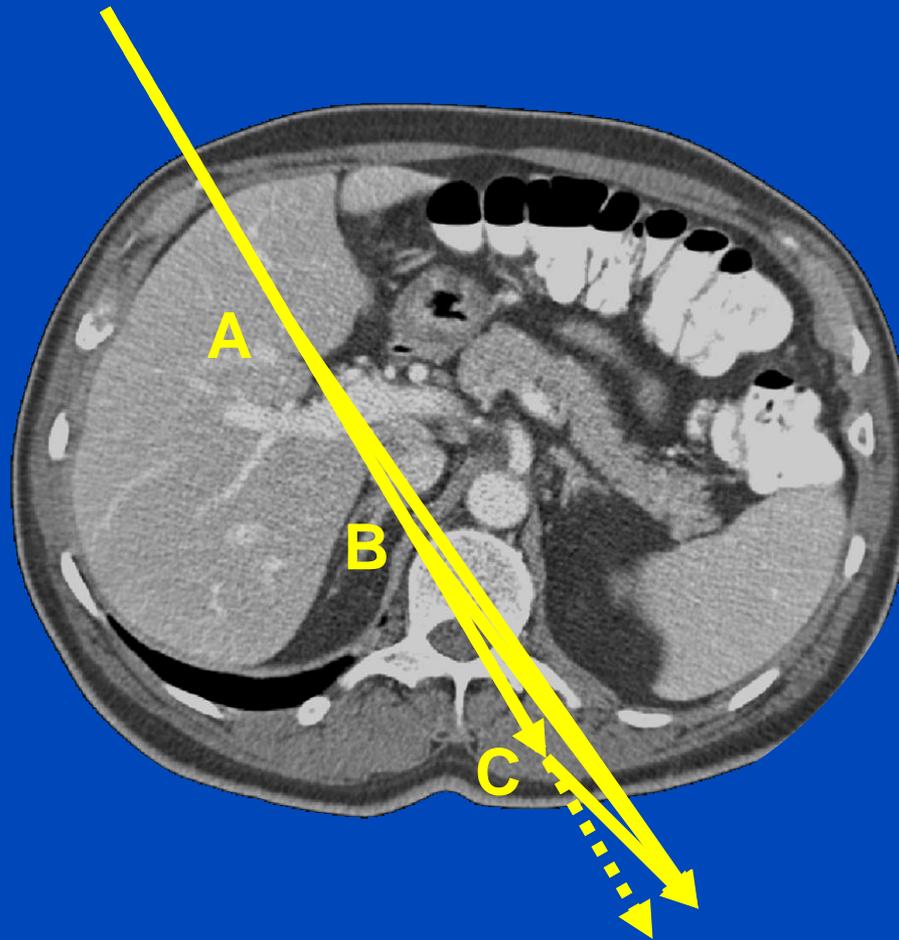
# PEP



# PEP



# PEP



$$(\mu(r_A) + \mu(r_B) + \mu(r_C))e^{-p} = p e^{-p}$$

# Ref 1: Kernel-Based Scatter Estimation

- Kernel-based scatter estimation<sup>1</sup>:

- Estimation of scatter by a convolution of the scatter source term  $T(p)$  with a scatter propagation kernel  $G(u, c)$ :

$$I_{s, \text{ est}}(\mathbf{u}) = \underbrace{\left( c_0 \cdot p(\mathbf{u}) \cdot e^{-p(\mathbf{u})} \right)}_{T(p)(\mathbf{u})} * \underbrace{\left( \sum_{\pm} e^{-c_1(u\hat{e}_1 \pm c_2)^2} \cdot \sum_{\pm} e^{-c_3(u\hat{e}_2 \pm c_4)^2} \right)}_{G(\mathbf{u}, \mathbf{c})}$$



$T(p)(\mathbf{u})$

Open parameters:  
 $c_0$



$G(\mathbf{u}, \mathbf{c})$

Open parameters:  
 $c_1, c_2, c_3, c_4$

$$\{c_i\} = \operatorname{argmin} \sum_n \sum_{\mathbf{u}} \|I_{s, \text{ est}}(n, \mathbf{u}, \{c_i\}) - I_s(n, \mathbf{u})\|_2^2,$$

Samples of the training data set

Scatter estimate

MC scatter simulation

Detector coordinate



<sup>1</sup> B. Ohnesorge, T. Flohr, K. Klingensbeck-Regn: Efficient object scatter correction algorithm for third and fourth generation CT scanners. Eur. Radiol. 9, 563–569 (1999).

# Ref 2: Hybrid Scatter Estimation

- Hybrid scatter estimation<sup>2</sup> :

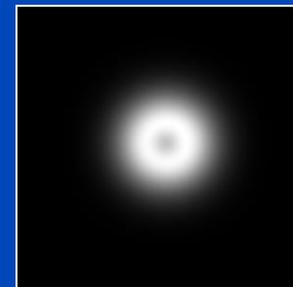
- Estimation of scatter by a convolution of the scatter source term  $T(p)$  with a scatter propagation kernel  $G(u, c)$ :

$$I_{s, \text{ est}}(\mathbf{u}) = \underbrace{\left( c_0 \cdot p(\mathbf{u}) \cdot e^{-p(\mathbf{u})} \right)}_{T(p)(\mathbf{u})} * \underbrace{\left( \sum_{\pm} e^{-c_1(\mathbf{u}\hat{\mathbf{e}}_1 \pm c_2)^2} \cdot \sum_{\pm} e^{-c_3(\mathbf{u}\hat{\mathbf{e}}_2 \pm c_4)^2} \right)}_{G(\mathbf{u}, \mathbf{c})}$$



$T(p)(\mathbf{u})$

Open parameters:  
 $c_0$



$G(\mathbf{u}, \mathbf{c})$

Open parameters:  
 $c_1, c_2, c_3, c_4$

$$\{c_i\}_n = \operatorname{argmin}_{\mathbf{u}} \sum \|I_{s, \text{ est}}(n, \mathbf{u}, \{c_i\}) - I_s(n, \mathbf{u})\|_2^2,$$

Samples of the test data set

Detector coordinate

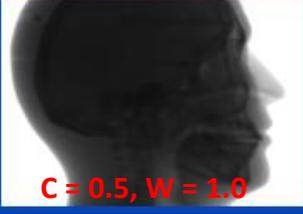
Scatter estimate



Coarse MC simulation



# Results on Simulated Projection Data

|         | Primary intensity   | Scatter ground truth (GT)   | (Kernel - GT) / GT  | (Hybrid - GT) / GT   | (DSE - GT) / GT  |
|---------|---|---|---|--|--|
| View #1 |    |    | <b>14.1%</b><br>mean absolute percentage error over all projections | <b>7.2%</b><br>mean absolute percentage error over all projections | <b>1.2%</b><br>mean absolute percentage error over all projections |
| View #2 |    |    |   |  |  |
| View #3 |    |    |   |  |  |
| View #4 |   |   |   |  |  |
| View #5 |  |  |   |  |  |
|         | <b>C = 0.5, W = 1.0</b>   | <b>C = 0.04, W = 0.04</b>   | <b>C = 0 %, W = 50 %</b>  | <b>C = 0 %, W = 50 %</b>   | <b>C = 0 %, W = 50 %</b>   |

DSE trained to estimate scatter from **primary plus scatter**: High accuracy

# Results on Simulated Projection Data

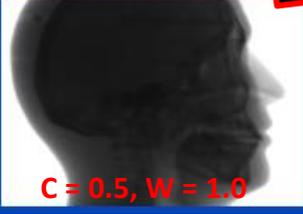
|         | Primary intensity   | Scatter ground truth (GT)   | (Kernel - GT) / GT  | (Hybrid - GT) / GT   | (DSE - GT) / GT  |
|---------|---|---|---|--|--|
| View #1 |    |    | <b>14.1%</b><br>mean absolute percentage error over all projections | <b>7.2%</b><br>mean absolute percentage error over all projections | <b>6.4%</b><br>mean absolute percentage error over all projections |
| View #2 |    |    |   |  |  |
| View #3 |    |    |   |  |  |
| View #4 |   |   |   |  |  |
| View #5 |  |  |   |  |  |

**DSE, in its present form, needs to see scatter in its input data!**

**C = 0.5, W = 1.0**      **C = 0.04, W = 0.04**      **C = 0 %, W = 50 %**      **C = 0 %, W = 50 %**      **C = 0 %, W = 50 %**

DSE trained to estimate scatter from **primary only**: Low accuracy

# Results on Simulated Projection Data

|         | Primary intensity   | Scatter ground truth (GT)   | (Kernel - GT) / GT  | (Hybrid - GT) / GT   | (DSE - GT) / GT  |
|---------|---|---|---|--|--|
| View #1 |    |    | <b>14.1%</b><br>mean absolute percentage error over all projections | <b>7.2%</b><br>mean absolute percentage error over all projections | <b>1.2%</b><br>mean absolute percentage error over all projections |
| View #2 |    |    |   |  |  |
| View #3 |    |    |   |  |  |
| View #4 |   |   |   |  |  |
| View #5 |  |  |   |  |  |

**DSE, in its present form, needs to see scatter in its input data!**

C = 0.5, W = 1.0

C = 0.04, W = 0.04

C = 0 %, W = 50 %

C = 0 %, W = 50 %

C = 0 %, W = 50 %

# Reconstructions of Simulated Data

Ground Truth

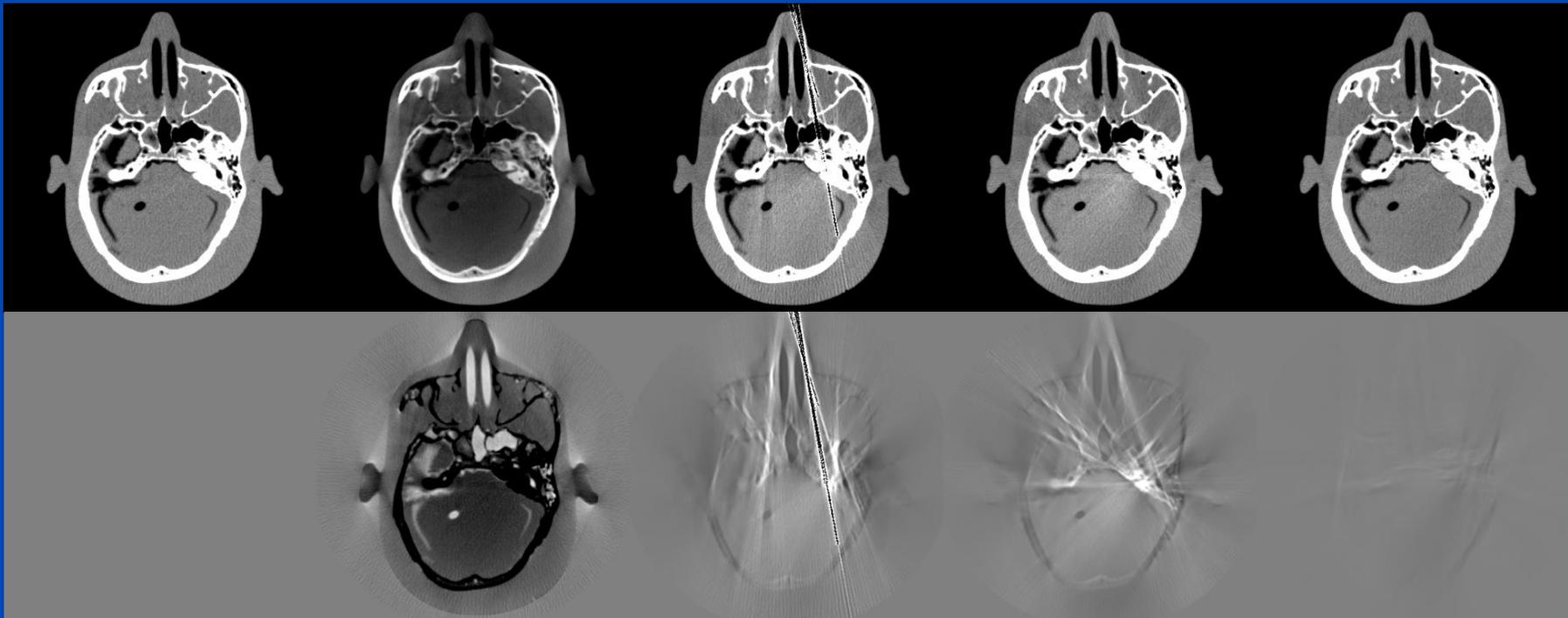
No Correction

Kernel-Based  
Scatter Estimation

Hybrid Scatter  
Estimation

Deep Scatter  
Estimation

CT Reconstruction  
Difference to ideal  
simulation



$C = 0 \text{ HU}, W = 1000 \text{ HU}$

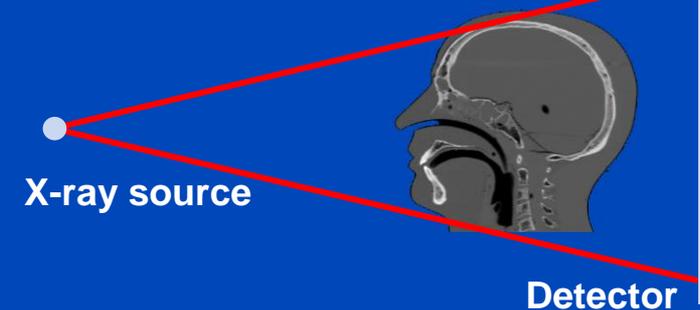
# Testing of the DSE Network for Measured Data (120 kV)

## DKFZ table-top CT

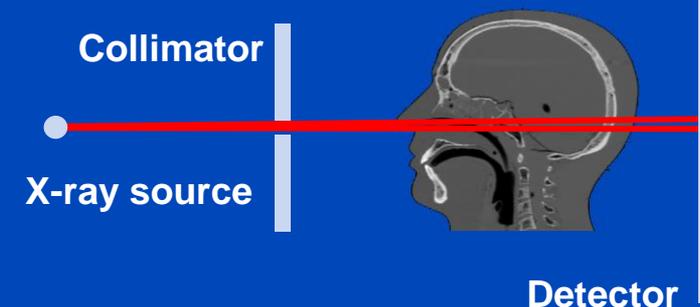


- Measurement of a head phantom at our in-house table-top CT.
- Slit scan measurement serves as ground truth.

### Measurement to be corrected



### Ground truth: slit scan



# Reconstructions of Measured Data

Slit Scan

No Correction

Kernel-Based  
Scatter Estimation

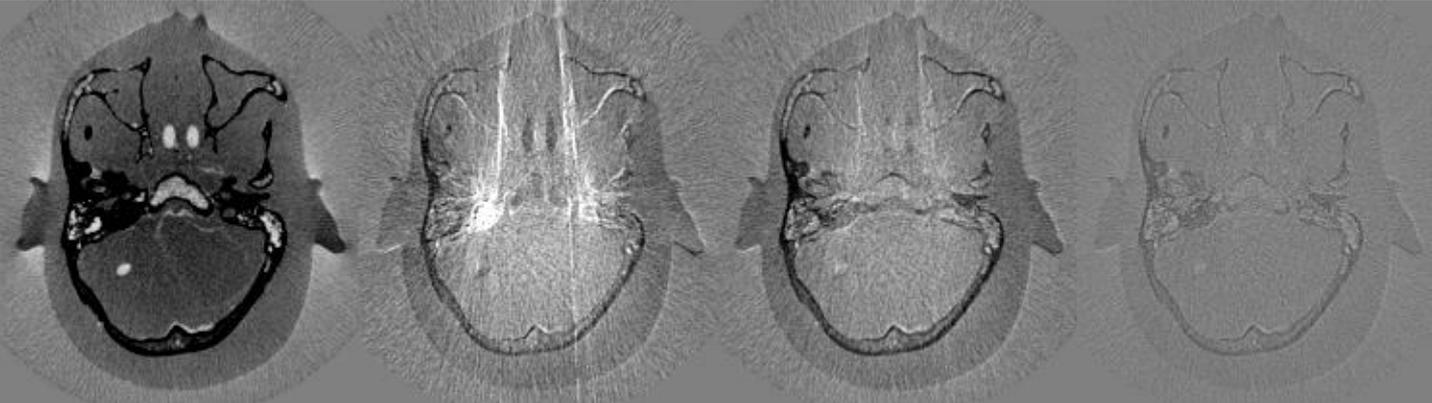
Hybrid Scatter  
Estimation

Deep Scatter  
Estimation

CT Reconstruction



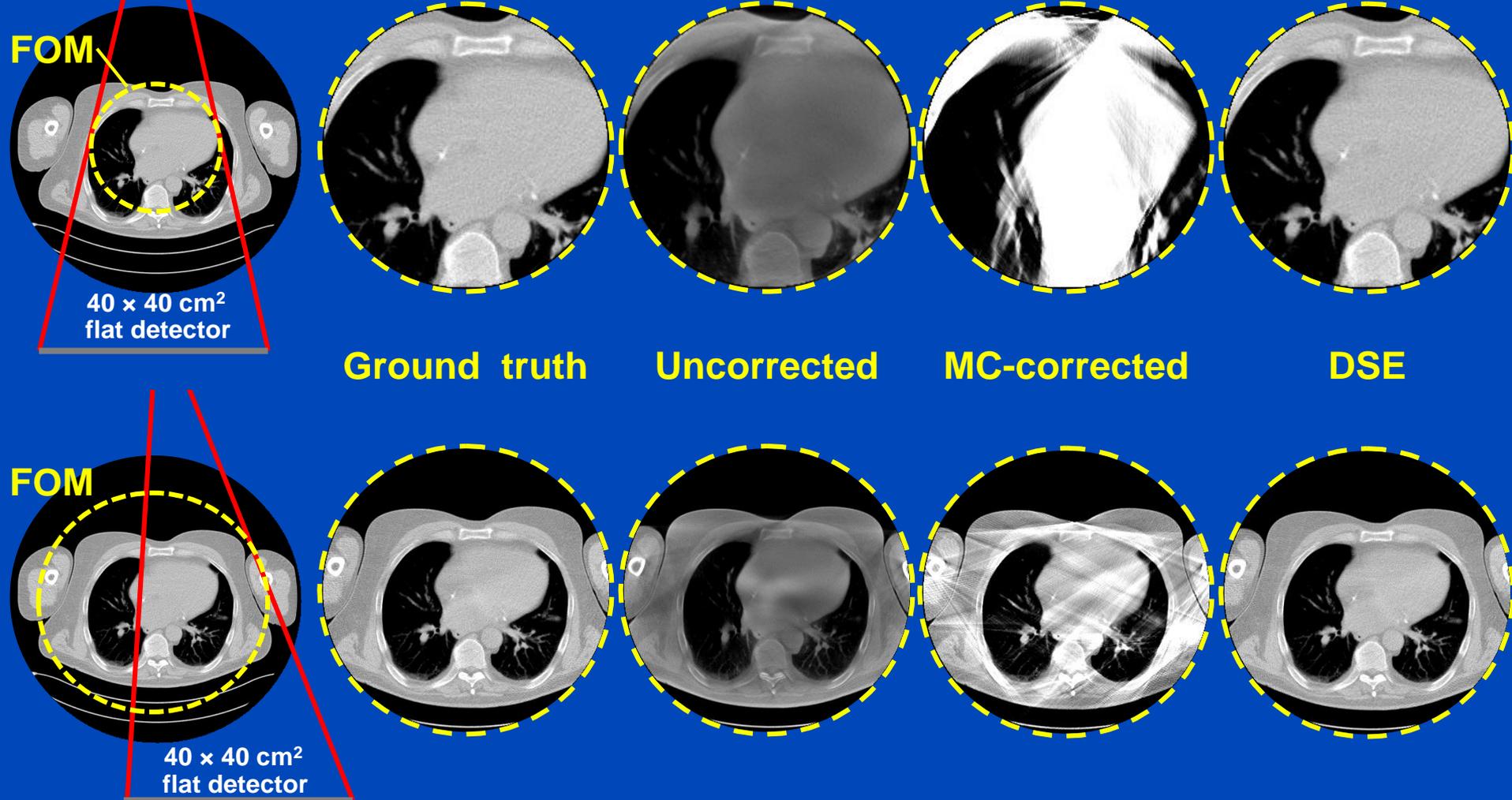
Difference to slit scan



$C = 0 \text{ HU}, W = 1000 \text{ HU}$

A simple detruncation was applied to the rawdata before reconstruction. Images were clipped to the FOM before display.  $C = -200$  HU,  $W = 1000$  HU.

# Truncated DSE<sup>1,2</sup>



To learn why MC fails at truncated data and what significant efforts are necessary to cope with that situation see [Kachelrieß et al. Effect of detruncation on the accuracy of MC-based scatter estimation in truncated CBCT. Med. Phys. 45(8):3574-3590, August 2018].

<sup>1</sup>J. Maier, M. Kachelrieß et al. Deep scatter estimation (DSE) for truncated cone-beam CT (CBCT). RSNA 2018.

<sup>2</sup>J. Maier, M. Kachelrieß et al. Robustness of DSE. Med. Phys. 46(1):238-249, January 2019.

# Generalization to Different Anatomical Regions

| <b>DSE</b> | Head       | Thorax     | Abdomen    |
|------------|------------|------------|------------|
| Head       | <b>1.2</b> | 21.1       | 32.7       |
| Thorax     | 8.8        | <b>1.5</b> | 9.1        |
| Abdomen    | 11.9       | 10.9       | <b>1.3</b> |
| All data   | <b>1.8</b> | <b>1.4</b> | <b>1.4</b> |

Values shown are the mean absolute percentage errors (MAPEs) of the testing data.  
Note that thorax and head suffer from truncation due to the small size of the 40×30 cm flat detector.

# Conclusions on DSE

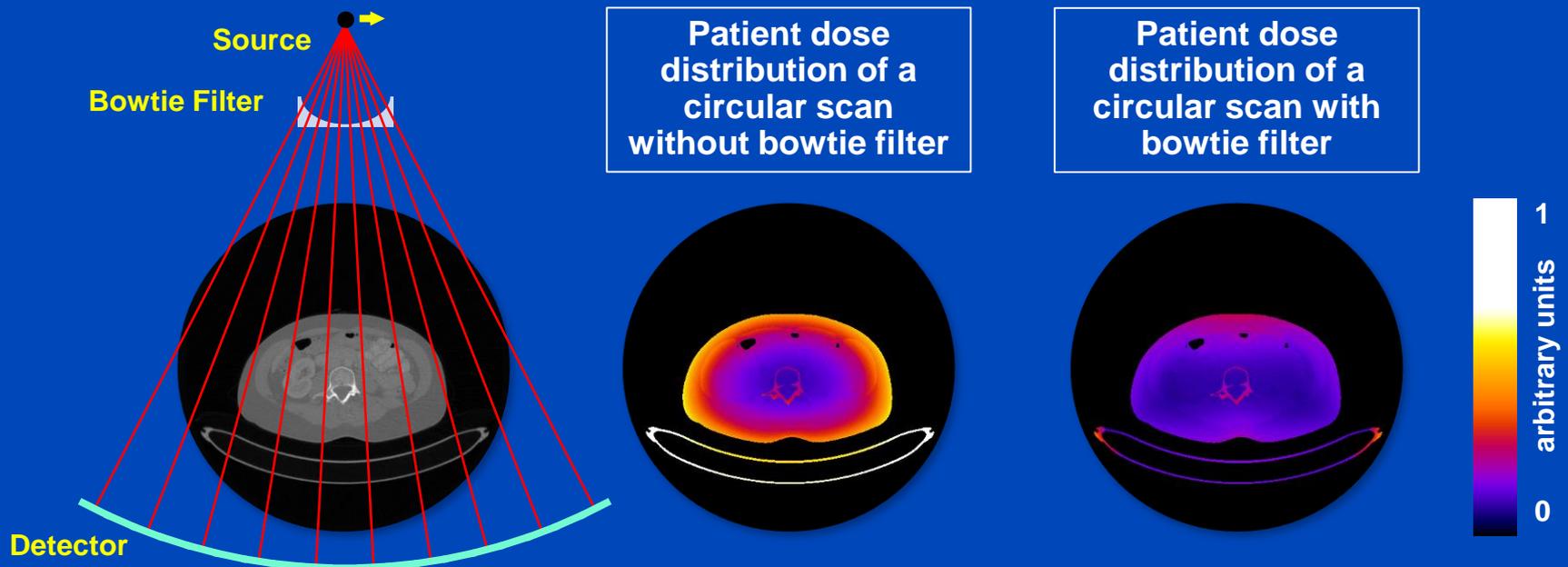
- DSE needs about 20 ms per projection. It is a fast and accurate alternative to Monte Carlo (MC) simulations.
- DSE outperforms kernel-based approaches in terms of accuracy and speed.
- Interesting observations
  - DSE can estimate scatter from a single (!) x-ray image.
  - DSE can accurately estimate scatter from a primary+scatter image.
  - DSE cannot accurately estimate scatter from a primary only image.
  - DSE may thus outperform MC even though DSE is trained with MC.
- DSE is not restricted to reproducing MC scatter estimates.
- DSE can rather be trained with any other scatter estimate, including those based on measurements.

# Estimation of Dose Distributions

- Useful to study dose reduction techniques
  - Tube current modulation
  - Prefiltration and shaped filtration
  - Tube voltage settings
  - ...
- Useful to estimate patient dose
  - Risk assessment requires segmentation of the organs (difficult)
  - Often semiantropomorphic patient models take over
  - The infamous k-factors that convert DLP into  $D_{\text{eff}}$  are derived this way, e.g.  $k_{\text{chest}} = 0.014 \text{ mSv/mGy/cm}$
  - ...
- Useful for patient-specific CT scan protocol optimization
- However: Dose estimation does not work in real time!

# Influence of Bowtie Filter

- Commercial CT-scanners are usually equipped with a bowtie filter in order to optimize the patient dose distribution.
- Monte-Carlo dose calculations or statistical reconstruction algorithms require exact knowledge of the bowtie filter.
- The shape as well as the composition of the bowtie filter is usually not disclosed by the CT vendors.



# Patient-Specific Dose Estimation

- **Accurate solutions:**
  - Monte Carlo (MC) simulation<sup>1</sup>, **gold standard**, stochastic LBTE solver
  - Analytic linear Boltzmann transport equation (LBTE) solver<sup>2</sup>
    - **Accurate but computationally expensive**
- **Fast alternatives:**
  - Application of patient-specific conversion factors to the DLP<sup>3</sup>.
  - Application of look-up tables using MC simulations of phantoms<sup>4</sup>.
  - Analytic approximation of CT dose deposition<sup>5</sup>.
    - **Fast but less accurate**

<sup>1</sup>G. Jarry et al., “A Monte Carlo-based method to estimate radiation dose from spiral CT”, Phys. Med. Biol. 48, 2003.

<sup>2</sup>A. Wang et al., “A fast, linear Boltzmann transport equation solver for computed tomography dose calculation (Acuros CTD)”. Med. Phys. 46(2), 2019.

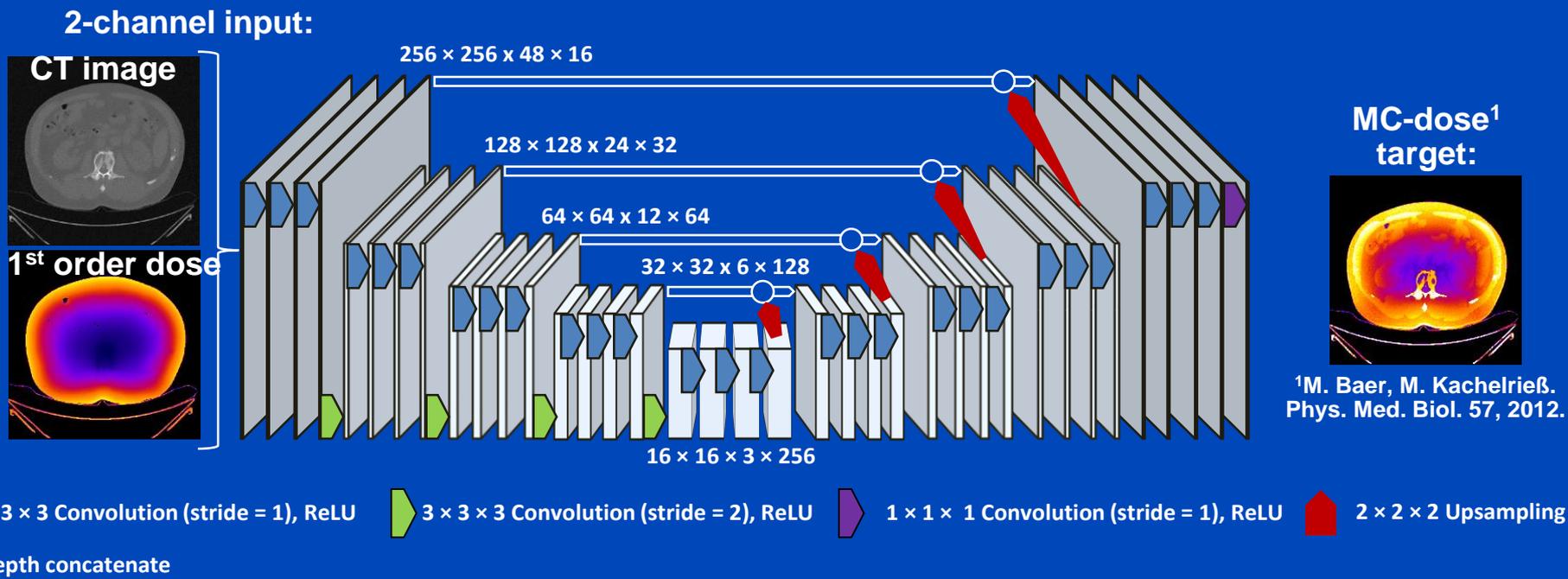
<sup>3</sup>B. Moore et al., “Size-specific dose estimate (SSDE) provides a simple method to calculate organ dose for pediatric CT examinations”, Med. Phys. 41, 2014.

<sup>4</sup>A. Ding et al., “VirtualDose: a software for reporting organ doses from CT for adult and pediatric patients”, Phys. Med. Biol. 60, 2015.

<sup>5</sup>B. De Man, “Dose reconstruction for real-time patient-specific dose estimation in CT”, Med. Phys. 42, 2015.

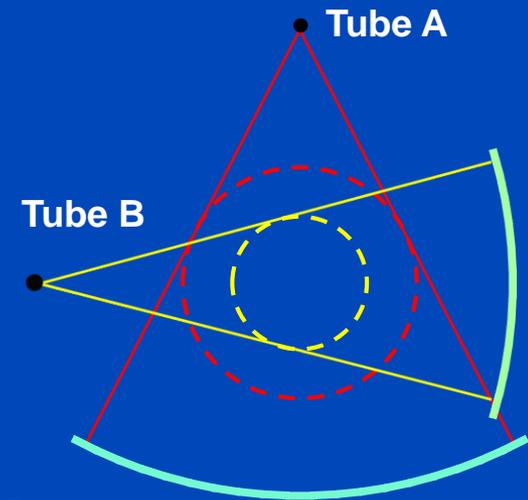
# Deep Dose Estimation (DDE)

- Combine fast and accurate CT dose estimation using a deep convolutional neural network.
- Train the network to reproduce MC dose estimates given the CT image and a first-order dose estimate.



# Training and Validation

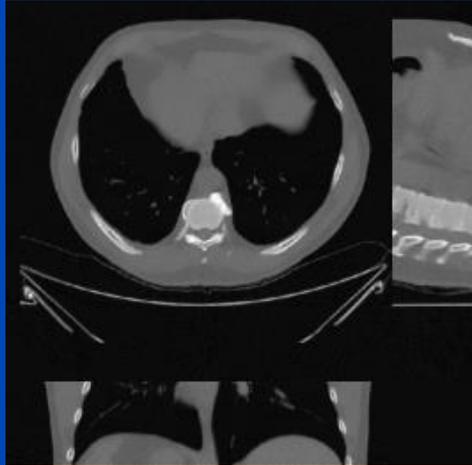
- Simulation of 1440 circular dual-source CT scans ( $64 \times 0.6$  mm,  $FOM_A = 50$  cm,  $FOM_B = 32$  cm) of thorax, abdomen, and pelvis using 12 different patients.
- Simulation with and without bowtie.
- No data augmentation
- Reconstruction on a  $512 \times 512 \times 96$  grid with 1 mm voxel size, followed by  $2 \times 2 \times 2$  binning for dose estimation.
- 9 patients were used for training and 3 for testing.
- DDE was trained for 300 epochs on an Nvidia Quadro P6000 GPU using a mean absolute error pixel-wise loss, the Adam optimizer, and a batch size of 4.
- The same weights and biases were used for all cases.



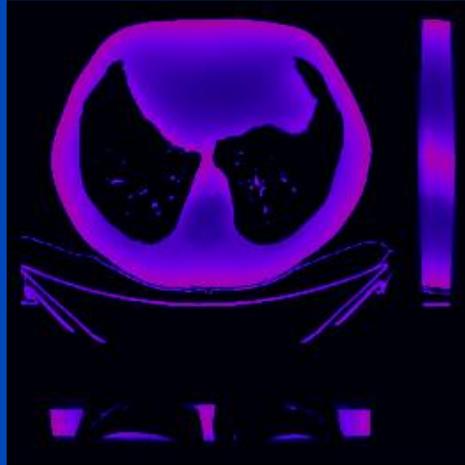
# Results

Thorax, tube A, 120 kV, with bowtie

CT image



First order dose

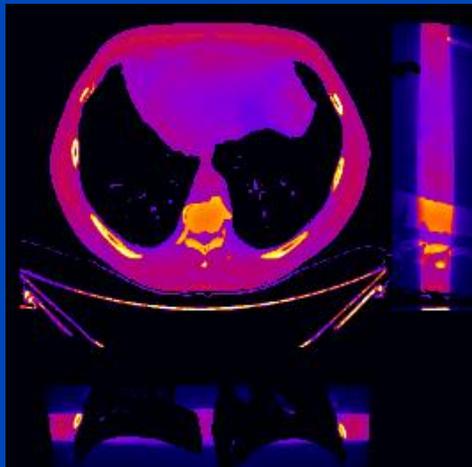


|            | MC   | DDE    |
|------------|------|--------|
| 48 slices  | 1 h  | 0.25 s |
| whole body | 20 h | 5 s    |

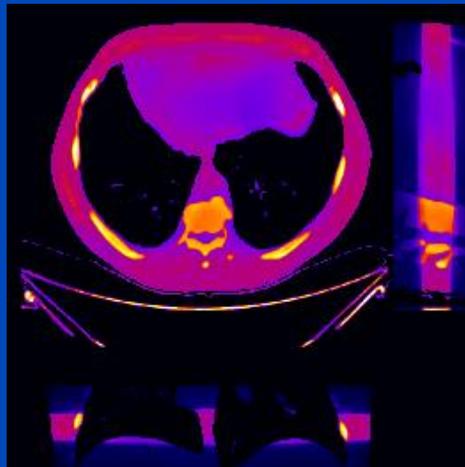
MC uses 16 CPU kernels  
DDE uses one Nvidia Quadro P600 GPU

DDE training took 74 h for 300 epochs,  
1440 samples, 48 slices per sample

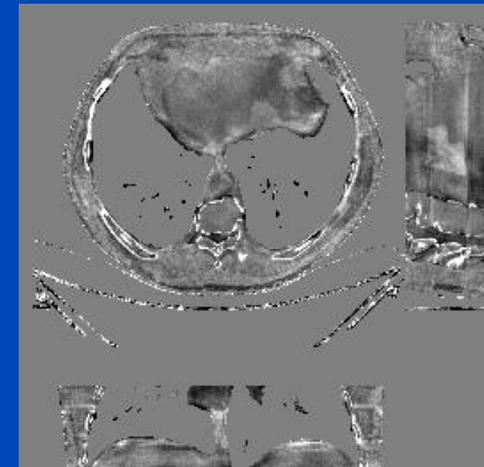
MC ground truth



DDE



Relative error

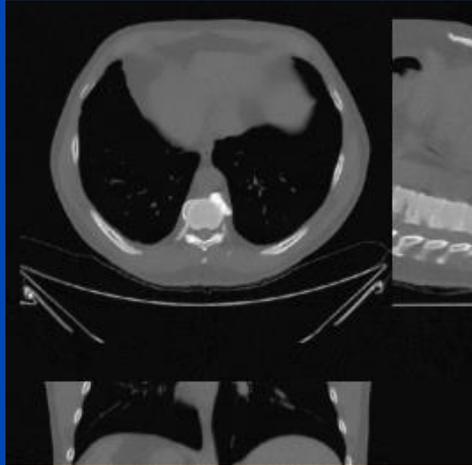


C = 0%  
W = 40%

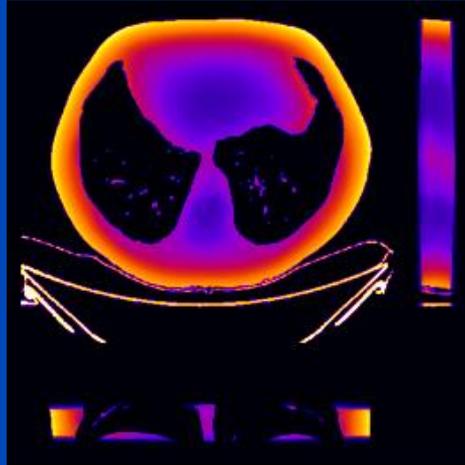
# Results

Thorax, tube A, 120 kV, no bowtie

CT image



First order dose

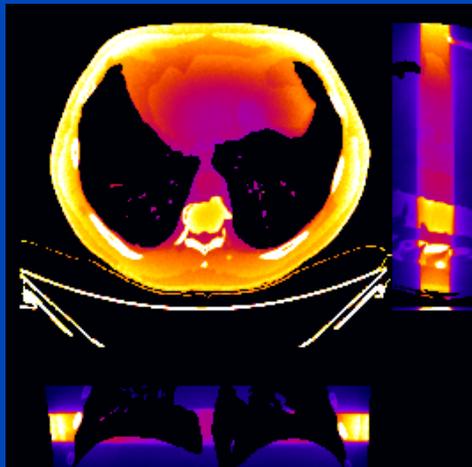


|            | MC   | DDE    |
|------------|------|--------|
| 48 slices  | 1 h  | 0.25 s |
| whole body | 20 h | 5 s    |

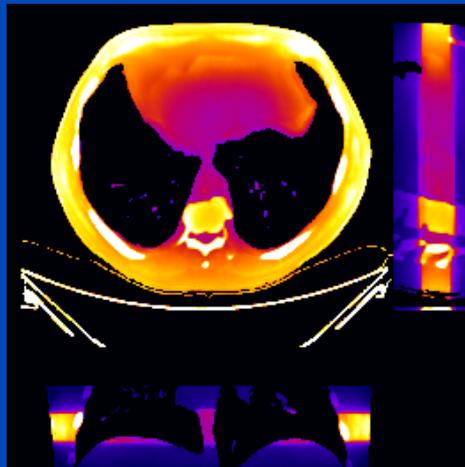
MC uses 16 CPU kernels  
DDE uses one Nvidia Quadro P600 GPU

DDE training took 74 h for 300 epochs,  
1440 samples, 48 slices per sample

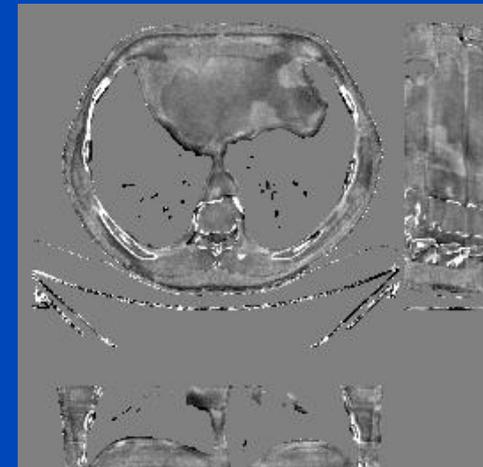
MC ground truth



DDE



Relative error

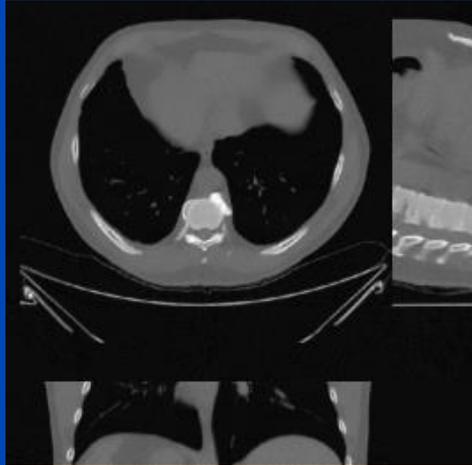


C = 0%  
W = 40%

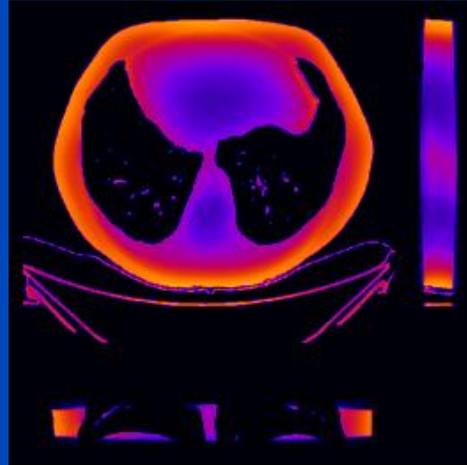
# Results

Thorax, tube B, 120 kV, no bowtie

CT image



First order dose

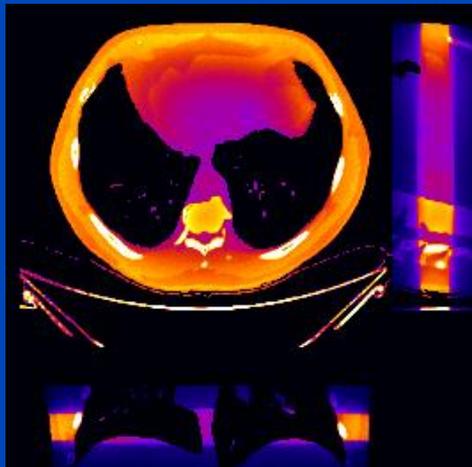


|            | MC   | DDE    |
|------------|------|--------|
| 48 slices  | 1 h  | 0.25 s |
| whole body | 20 h | 5 s    |

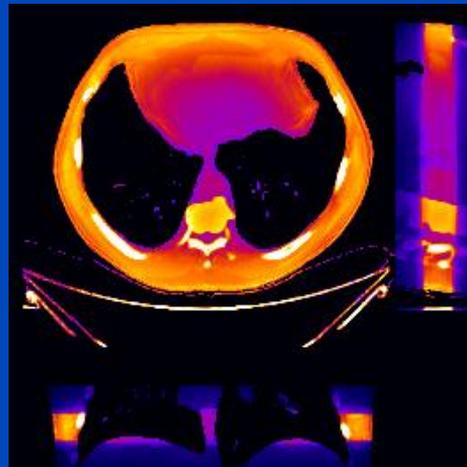
MC uses 16 CPU kernels  
DDE uses one Nvidia Quadro P600 GPU

DDE training took 74 h for 300 epochs,  
1440 samples, 48 slices per sample

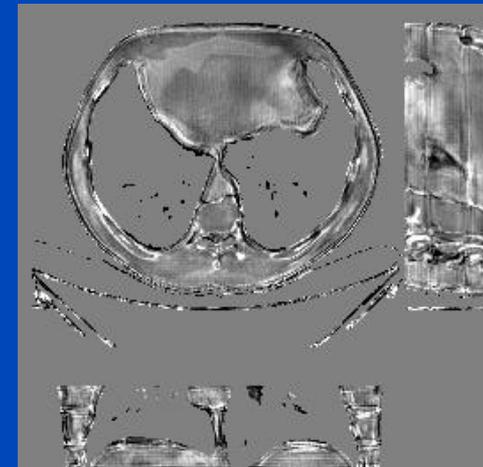
MC ground truth



DDE



Relative error

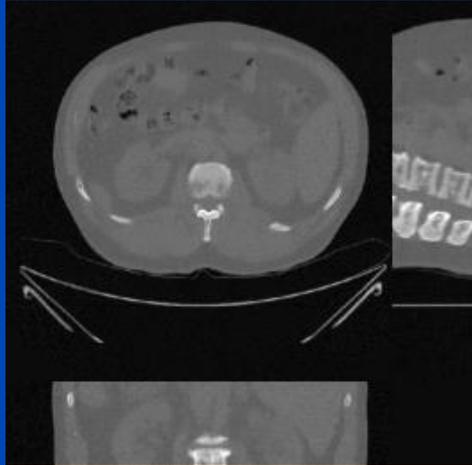


C = 0%  
W = 40%

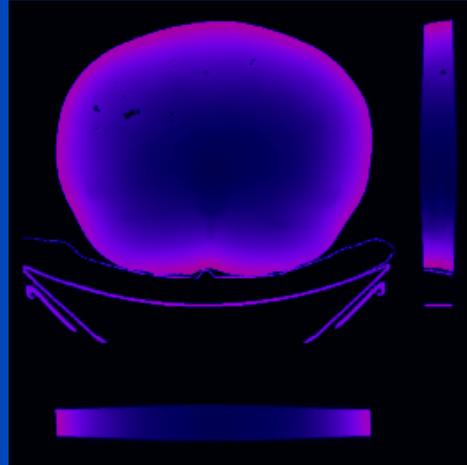
# Results

Abdomen, tube A, 120 kV, with bowtie

CT image



First order dose

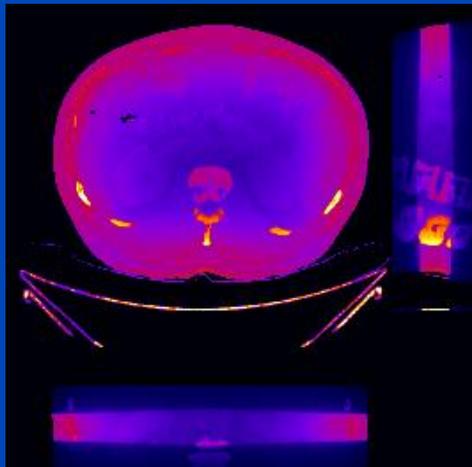


|            | MC   | DDE           |
|------------|------|---------------|
| 48 slices  | 1 h  | <b>0.25 s</b> |
| whole body | 20 h | <b>5 s</b>    |

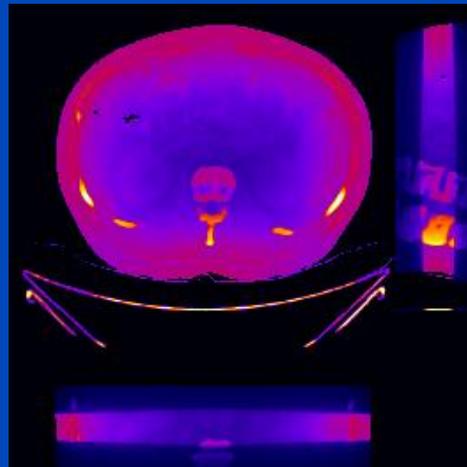
MC uses 16 CPU kernels  
DDE uses one Nvidia Quadro P600 GPU

DDE training took 74 h for 300 epochs,  
1440 samples, 48 slices per sample

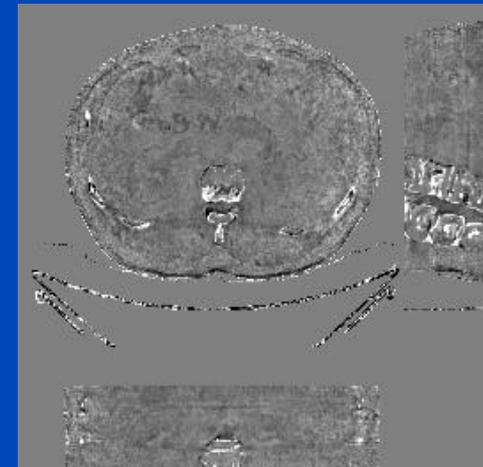
MC ground truth



DDE



Relative error

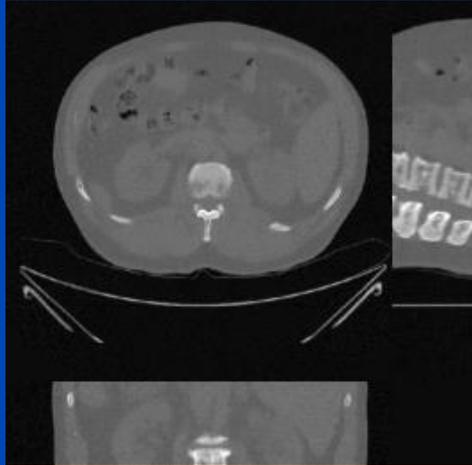


C = 0%  
W = 40%

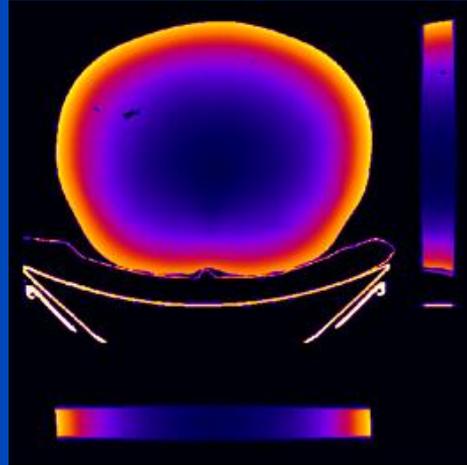
# Results

Abdomen, tube A, 120 kV, no bowtie

CT image



First order dose

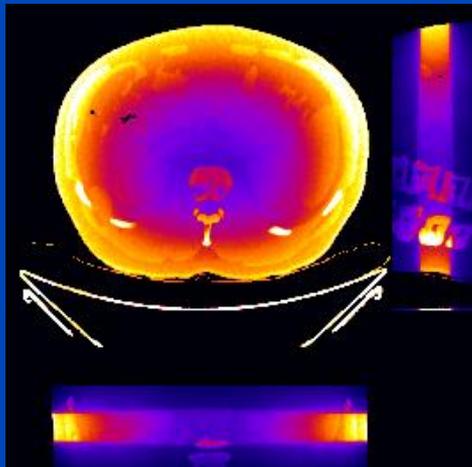


|            | MC   | DDE           |
|------------|------|---------------|
| 48 slices  | 1 h  | <b>0.25 s</b> |
| whole body | 20 h | <b>5 s</b>    |

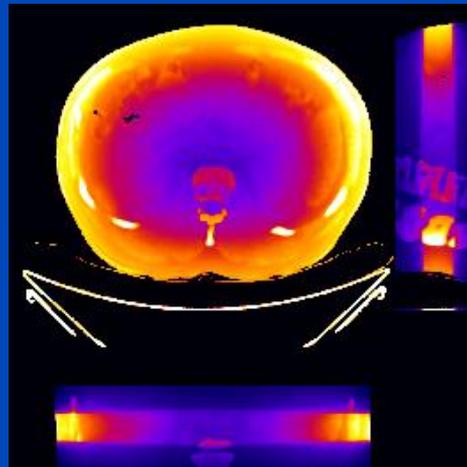
MC uses 16 CPU kernels  
DDE uses one Nvidia Quadro P600 GPU

DDE training took 74 h for 300 epochs,  
1440 samples, 48 slices per sample

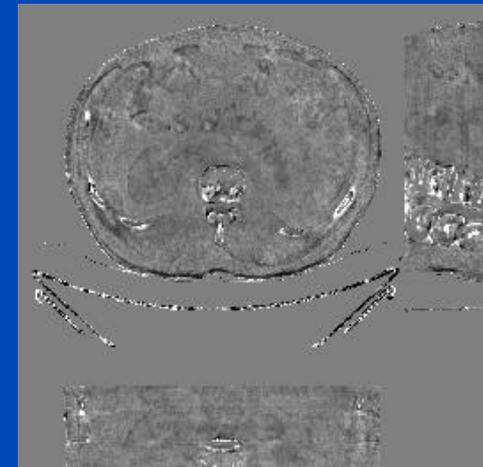
MC ground truth



DDE



Relative error

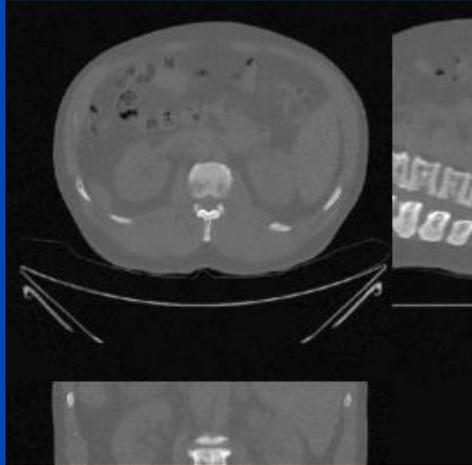


C = 0%  
W = 40%

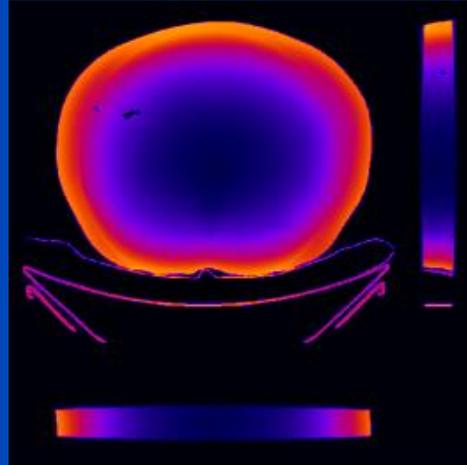
# Results

Abdomen, tube B, 120 kV, no bowtie

CT image



First order dose

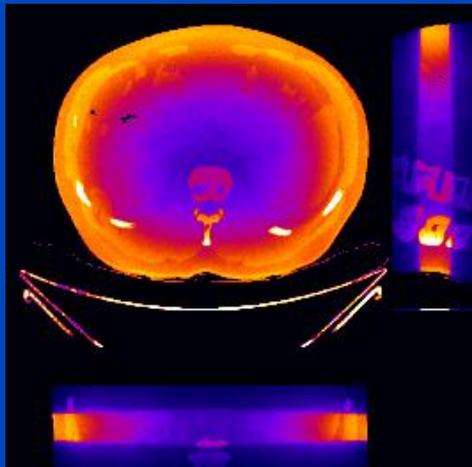


|            | MC   | DDE    |
|------------|------|--------|
| 48 slices  | 1 h  | 0.25 s |
| whole body | 20 h | 5 s    |

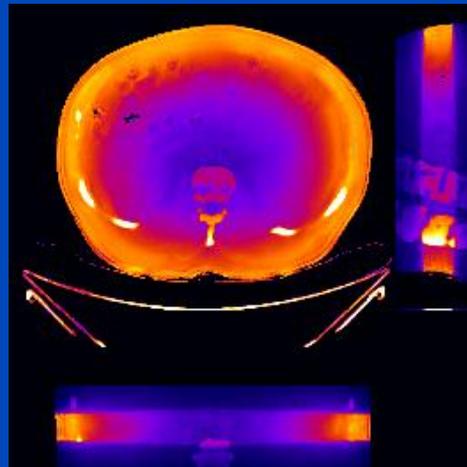
MC uses 16 CPU kernels  
DDE uses one Nvidia Quadro P600 GPU

DDE training took 74 h for 300 epochs,  
1440 samples, 48 slices per sample

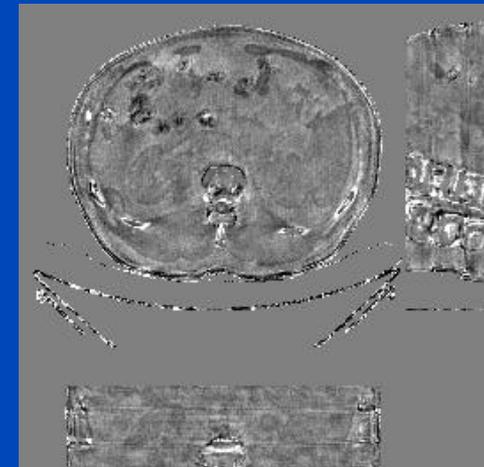
MC ground truth



DDE



Relative error

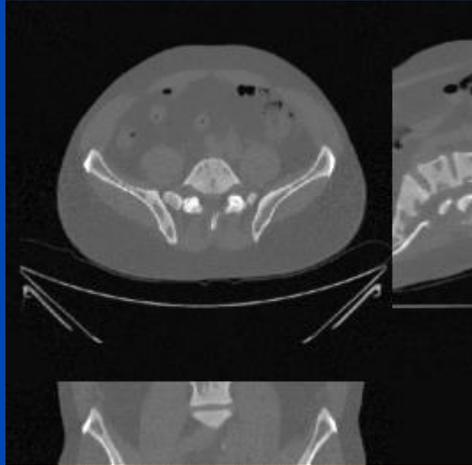


C = 0%  
W = 40%

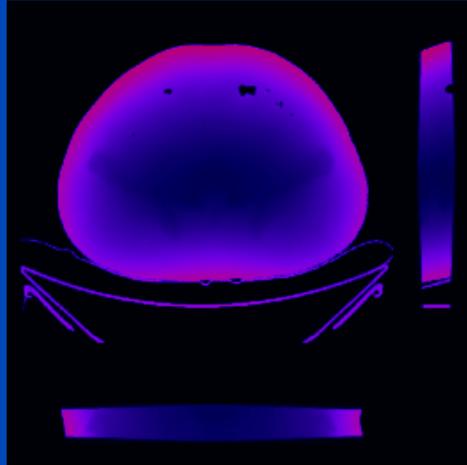
# Results

Pelvis, tube A, 120 kV, with bowtie

CT image



First order dose

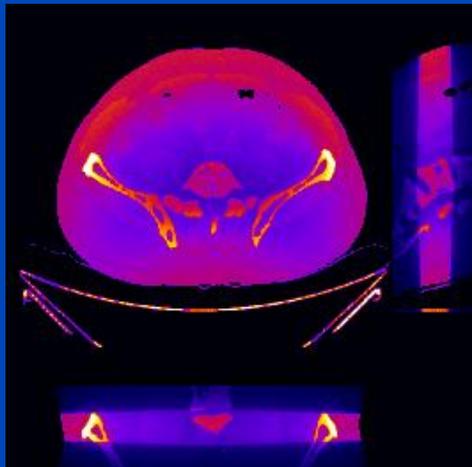


|            | MC   | DDE           |
|------------|------|---------------|
| 48 slices  | 1 h  | <b>0.25 s</b> |
| whole body | 20 h | <b>5 s</b>    |

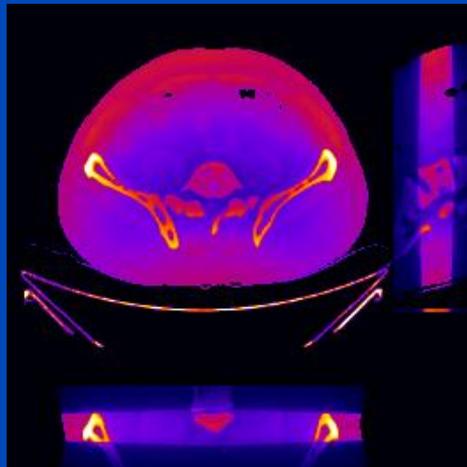
MC uses 16 CPU kernels  
DDE uses one Nvidia Quadro P600 GPU

DDE training took 74 h for 300 epochs,  
1440 samples, 48 slices per sample

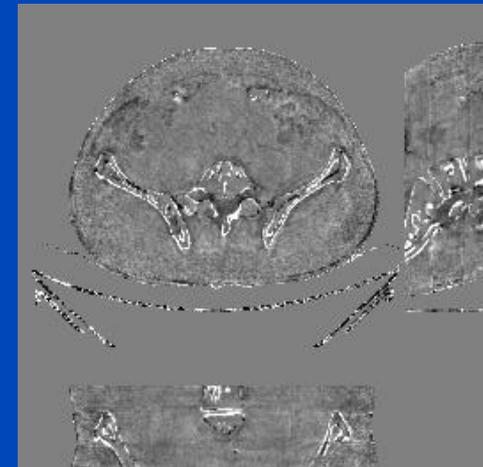
MC ground truth



DDE



Relative error

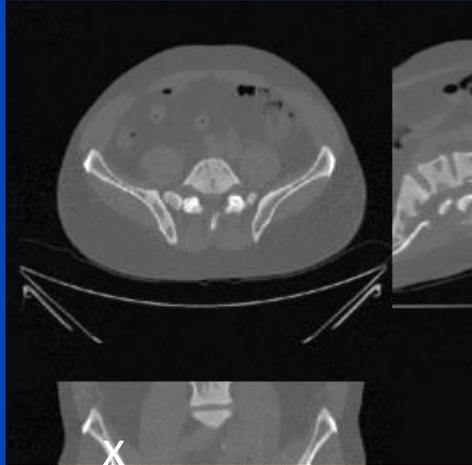


**C = 0%**  
**W = 40%**

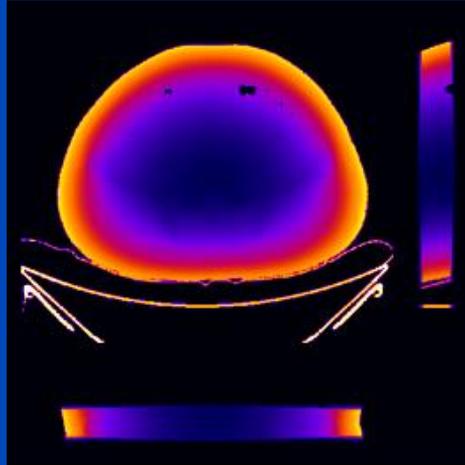
# Results

Pelvis, tube A, 120 kV, no bowtie

CT image



First order dose

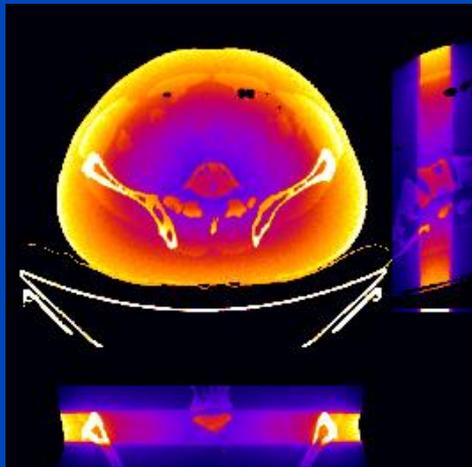


|            | MC   | DDE           |
|------------|------|---------------|
| 48 slices  | 1 h  | <b>0.25 s</b> |
| whole body | 20 h | <b>5 s</b>    |

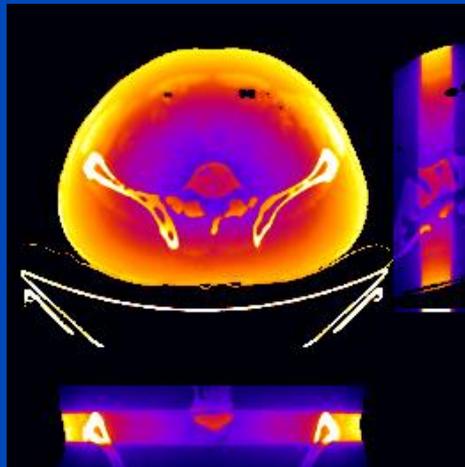
MC uses 16 CPU kernels  
DDE uses one Nvidia Quadro P600 GPU

DDE training took 74 h for 300 epochs,  
1440 samples, 48 slices per sample

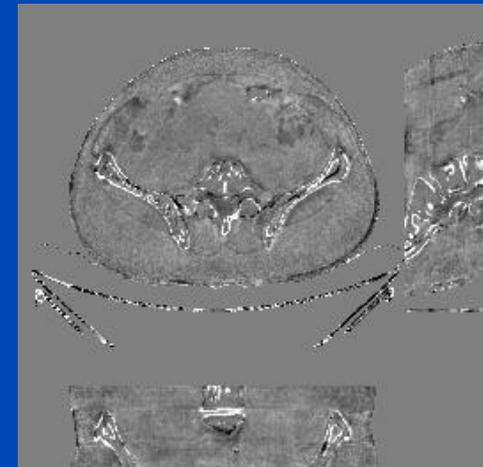
MC ground truth



DDE



Relative error

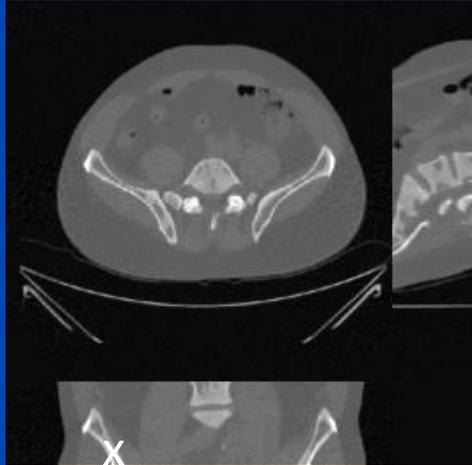


**C = 0%**  
**W = 40%**

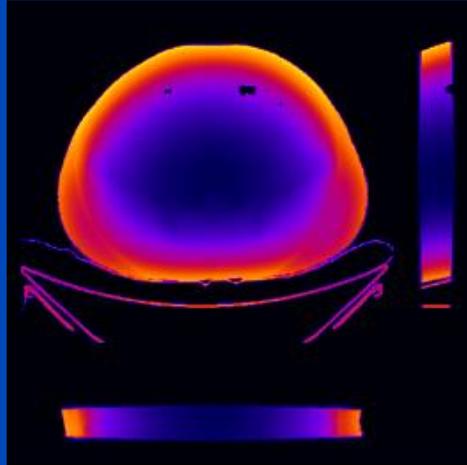
# Results

Pelvis, tube B, 120 kV, no bowtie

CT image



First order dose

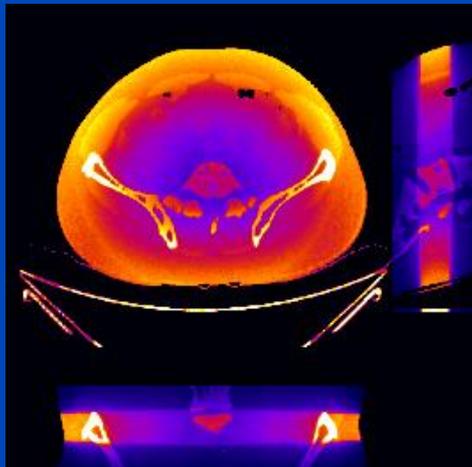


|            | MC   | DDE    |
|------------|------|--------|
| 48 slices  | 1 h  | 0.25 s |
| whole body | 20 h | 5 s    |

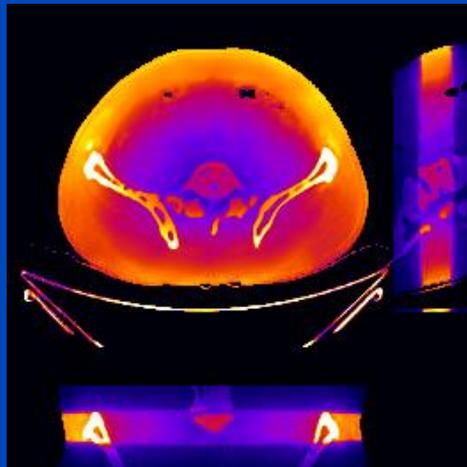
MC uses 16 CPU kernels  
DDE uses one Nvidia Quadro P600 GPU

DDE training took 74 h for 300 epochs,  
1440 samples, 48 slices per sample

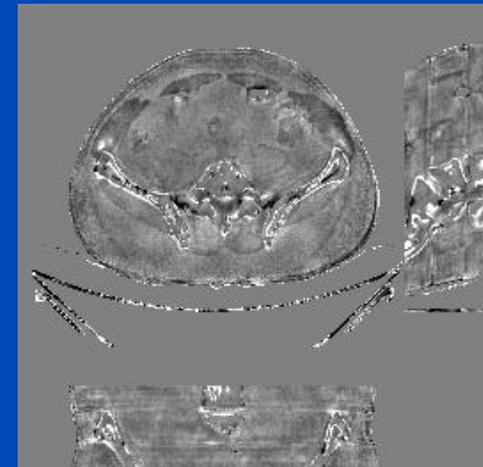
MC ground truth



DDE



Relative error



C = 0%  
W = 40%

# Conclusions on DDE

- As shown, DDE works well with 360° circle scans.
- What is not shown in this presentation is that DDE can be trained to provide accurate dose predictions
  - for sequence scans
  - for partial scans (less than 360°)
  - for spiral scans
  - for different tube voltages
  - for scans with and without bowtie filtration
  - for scans with tube current modulation
- In practice it may therefore be not necessary to perform separate training runs for these cases.
- Thus, accurate real-time patient dose estimation may become feasible with DDE.

# Part 4:

# Image Reconstruction

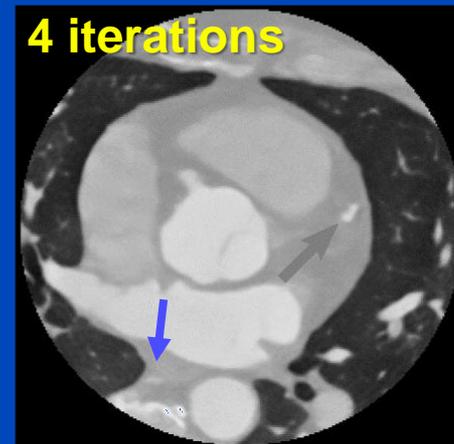
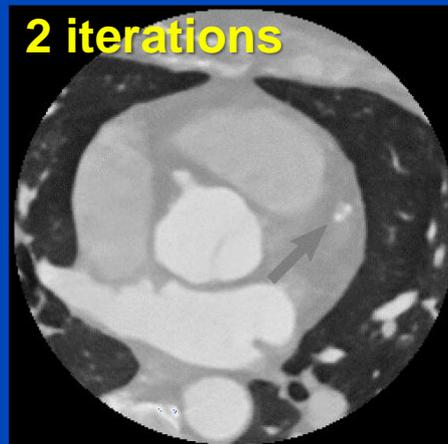
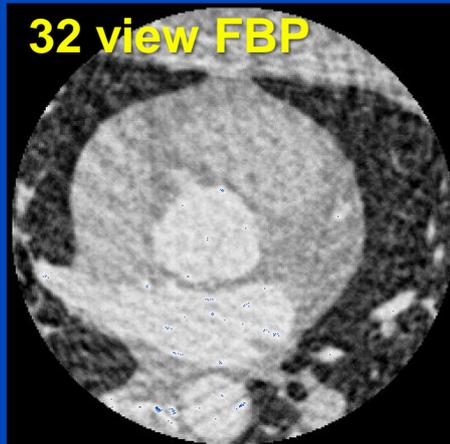
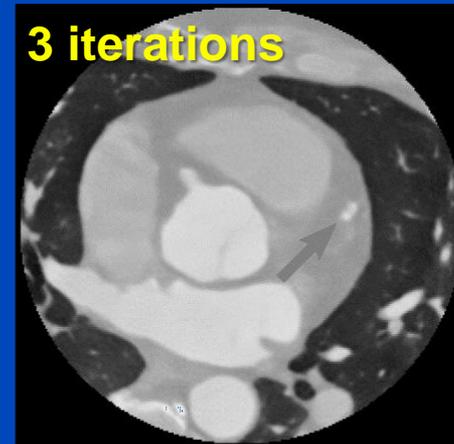
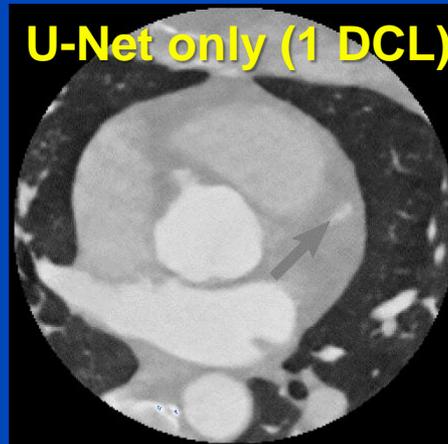
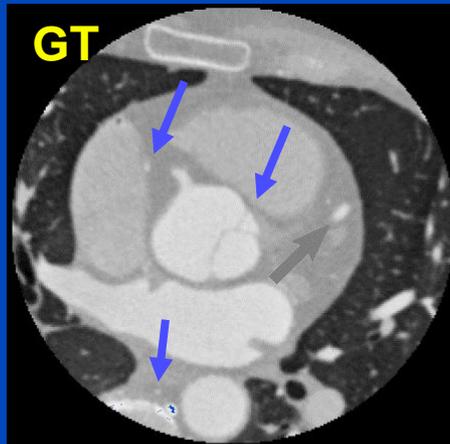
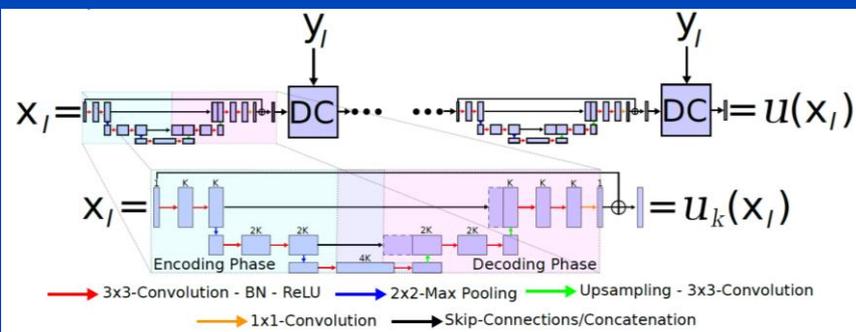
# Often “Just” Image Restoration

- **Speeding up iterative reconstruction by training a CNN to convert an FBP image into an iterative image**
  - Canon’s AiCE algorithm
  - GE’s True Fidelity algorithm
  - plus a few more algorithms proposed in the literature
- **Noise reduction by training, e.g. a mapping from low dose to high dose images**
  - many examples in the literature, some in this presentation
- **Artifact reduction in image domain**
  - many examples in the literature, one shown in this presentation
- ...

# Sometimes “Real” Image Reconstruction

- Networks employing data consistency layers
- Networks including backprojection layers
- Learning of backprojectors
- End-to-end training from sinogram to image
- Unrolled iterative reconstruction with learned priors
- ...

# Sparse CT Recon with Data Consistency Layers (DCLs)



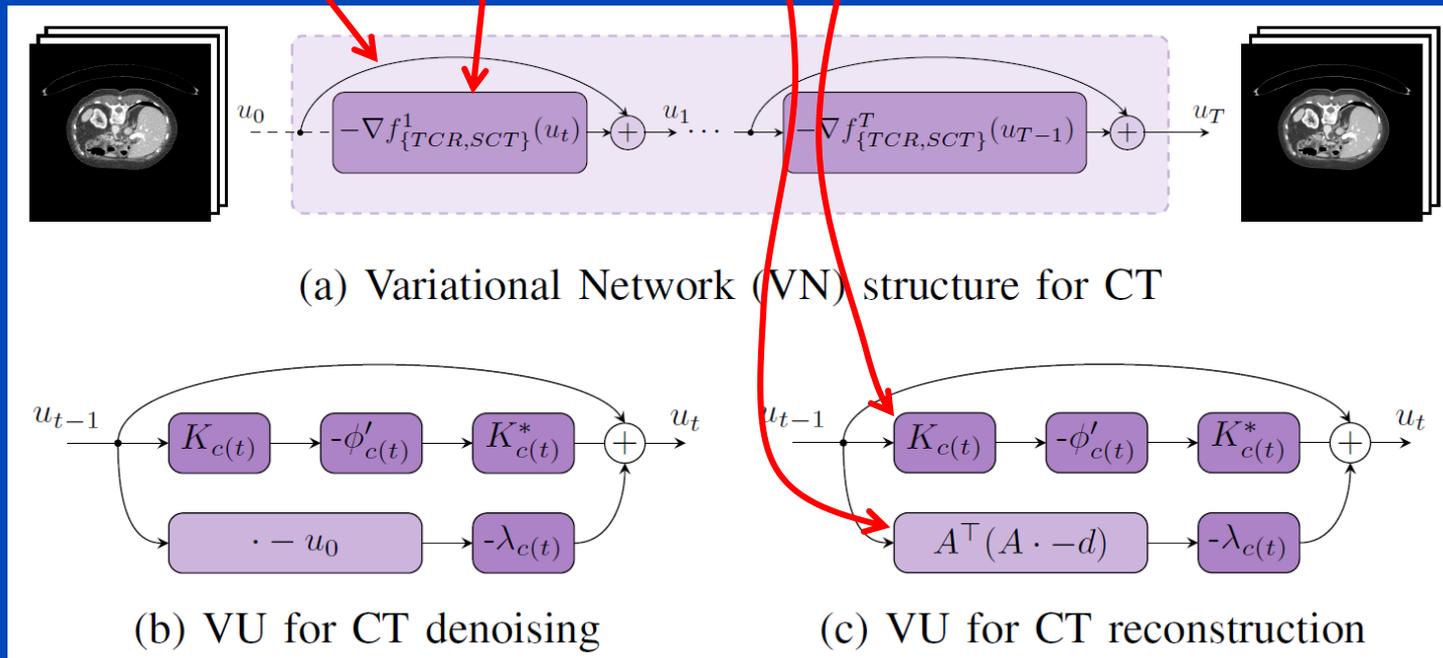
# Variational Network-Based Image Reconstruction

$$C(f) = \|X \cdot f - p\|_W^2 + R(f)$$

$$\nabla C(f) = X^T \cdot W \cdot (X \cdot f - p) + \nabla R(f)$$

$$f^{(t+1)} = f^{(t)} - \lambda \nabla C(f^{(t)})$$

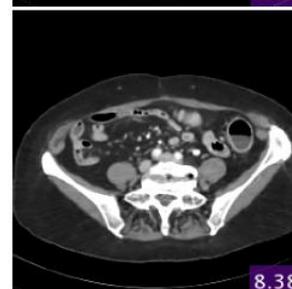
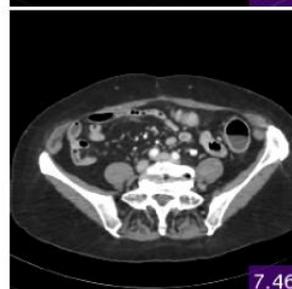
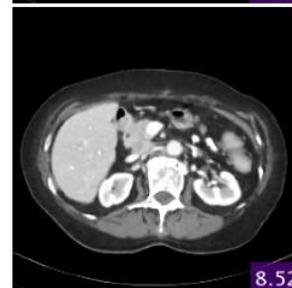
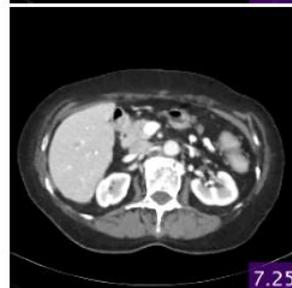
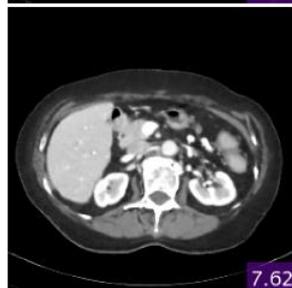
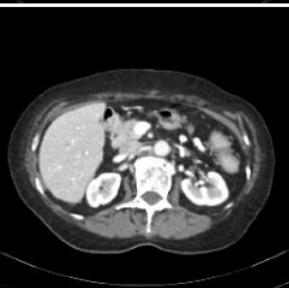
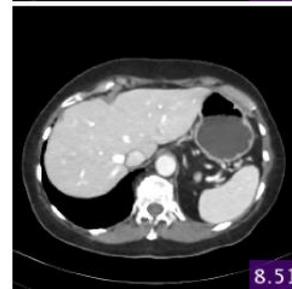
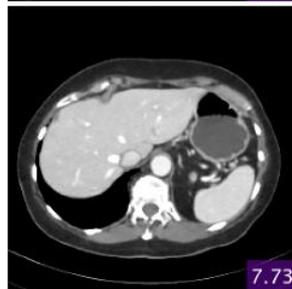
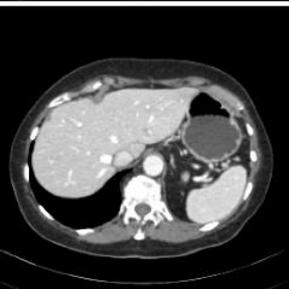
Highly simplified example. Varnets work for a much wider class of cost functions whose NN-based minimization is motivated by the primal dual approach.



full dose

1/4 dose

1/6 dose



(a) full-dose

(b) SAFIRE

(c) TV

(d) TCR

(e) SCT

(f) SCT

tube current reduction  
SAFIRE

sparse views  
TV

tube current reduction  
varnet

sparse views  
varnet

sparse views  
varnet

# Conclusions on Deep CT

- Machine learning will play a significant role in CT image formation.
- High potential for
  - Artifact correction
  - Noise and dose reduction
  - Real-time dose assessment (also for RT)
  - ...
- Care has to be taken
  - Underdetermined acquisition, e.g. sparse view or limited angle CT, require the net to make up information!
  - Nice looking images do not necessarily represent the ground truth.
  - Data consistency layers and variational networks with rawdata access may ensure that the information that is made up is consistent with the measured data.
  - ...



# Thank You!



## The 6<sup>th</sup> International Conference on Image Formation in X-Ray Computed Tomography

August 3 - August 7 • 2020 • Regensburg • Germany • [www.ct-meeting.org](http://www.ct-meeting.org)



Conference Chair: **Marc Kachelrieß**, German Cancer Research Center (DKFZ), Heidelberg, Germany

This presentation will soon be available at [www.dkfz.de/ct](http://www.dkfz.de/ct).  
Job opportunities through DKFZ's international Fellowship programs ([marc.kachelriess@dkfz.de](mailto:marc.kachelriess@dkfz.de)).  
Parts of the reconstruction software were provided by RayConStruct® GmbH, Nürnberg, Germany.